Software Risk Evaluation (SRE) Team Member's Notebook (Version 2.0)

Original Development

George J. Pandelios
Sandra G. Behrens
Richard L. Murphy
Ray C. Williams
William R. Wilson

Version 2.0 Revisions

Ray C. Williams

December 1999

TECHNICAL REPORT
CMU/SEI-99-TR-029
ESC-TR-99-029

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Software Risk Evaluation (SRE) Team Member's Notebook (Version 2.0)

CMU/SEI-99-TR-029
ESC-TR-99-029

Original Development

George J. Pandelios
Sandra G. Behrens
Richard L. Murphy
Ray C. Williams
William R. Wilson

Version 2.0 Revisions

Ray C. Williams

December 1999

Process Improvement Team

Software Engineering Process Management

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Norton L. Compton, Lt Col., USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

Abstract

The Software Risk Evaluation (SRE) is a process for identifying, analyzing, and developing mitigation strategies for risks in a software-intensive system while it is in development. The SRE process has been in evolutionary development at the SEI since 1992 and has been used on over 50 Department of Defense (DoD) and civil (federal and state) contractors and program offices.

The SRE Team Member's Notebook was written for the SEI's own use in administering SREs. It is a "prescriptive" document—long on direction and short on explanation. It is being published as an appendix to SRE Method Description Version 2.0 to provide an example of a specific procedure that complies with the SRE Method Description. Because the size and life-cycle duration of individual projects may vary widely, the SRE Team Member's Notebook may not be ideal for all organizations. It is intended as a starting point for organizations to create a similar document that meets their unique needs.

Appendix
SRE Team Member's Notebook (Version 2.0)

To The Reader

Document Purpose

This Software Risk Evaluation (SRE) Team Member's Notebook (TMNB) is a dual-purpose document. The two purposes are

1. as an appendix to the Method Description, Version 2.0, to provide an example of a straightforward process flow description
2. as a stand-alone document, to be carried by each SEI SRE team member and *used* (marked on, flagged, highlighted, torn apart) in the course of the SRE

The first purpose can be met with a static, unchanging document as you see here. The second requires that there will be revisions as the SEI performs SREs over time—there could be a revision per SRE, depending on how strictly future SEI team leaders feel bound to follow (and revise) the processes as written.

Intended Audience

This TMNB is written for use by SEI SRE team leaders and team members, and by SRE team members drawn from the local organization as part of an SEI-led SRE.

Relationship to the Method Description

This document only provides detail information on three phases outlined in the SRE Method Description:

- the *Risk Identification and Analysis* (RI&A) phase,
- the *Interim Report* phase
- the *Mitigation Strategy Planning* (MSP) phase

The *Contracting* and *Final Report* phases are considered to be primarily the SRE team leader's responsibility (with little or no involvement from

the rest of the team) and are described adequately in the SRE Method Description.

This TMNB does not attempt to explain why the particular tools and methods were selected to implement the SRE process. In many cases, this was an arbitrary choice of the SEI person responsible for that step; if it seemed to work, it stayed in. In other cases, the tool or method was chosen after another was tried and abandoned. The TMNB does not provide historical insight—just tools and methods that have been tried successfully by the SEI.

High-Level Content Description

This TMNB contains the following:

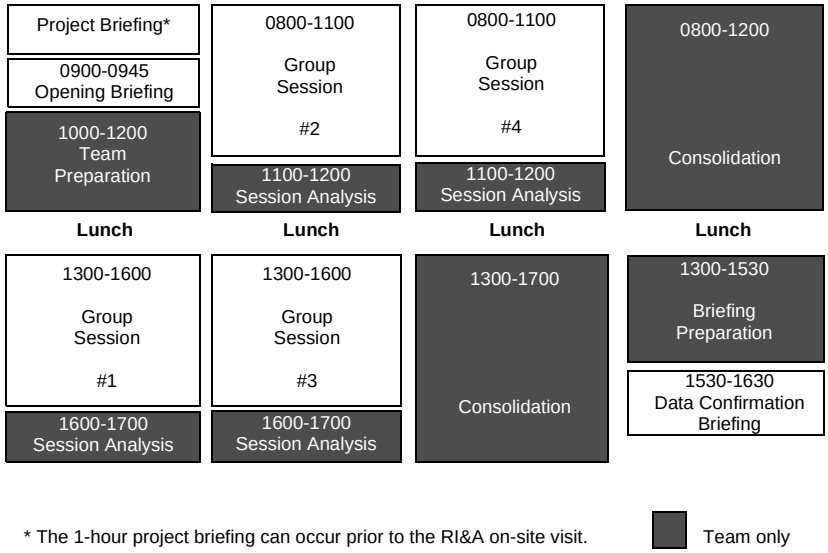
- summaries of all SRE activities during the RI&A, Interim Report, and MSP phases, much of which is carried out on-site in the offices of the project receiving the SRE
- checklists and sample forms used throughout the process

Document Organization

The TMNB is organized according to the schedule of activities during a "typical" SRE conducted by the SEI. Note that the MSP phase is often delayed by weeks or months after the completion of the Interim Report.

The TMNB is structured first in the order of the three phases depicted below, and within the phases in the order that a *unique* block occurs temporally.

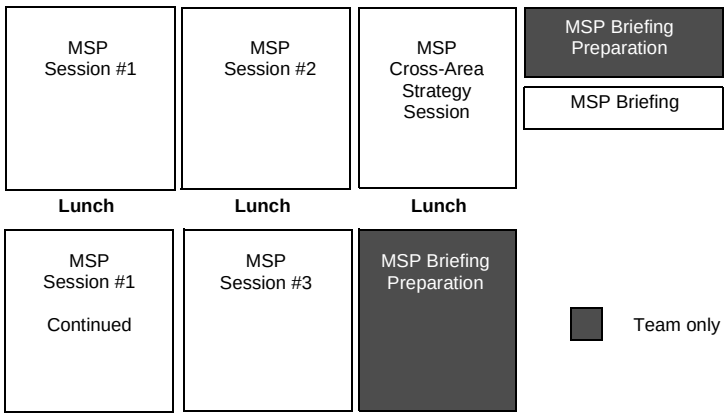
Risk Identification and Analysis (RI&A) Phase



Interim Report Phase*

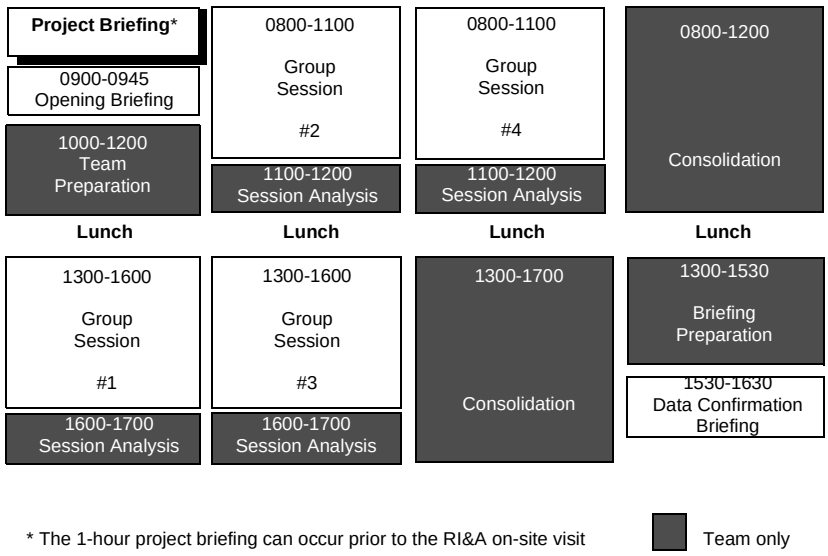
* The Interim Report Phase should begin immediately after the RI&A phase and be completed within two weeks.

Mitigation Strategy Planning Phase



MSP Sessions may range from 1/2 to 1 day in length, depending on the size of the risk area and the order of sessions. The first session will take the longest.

Project Briefing



Objectives to provide context and background to the SRE team on the project

Who's in the Room?

- the project manager (or designated substitute) who is giving the briefing
- any other project members the project manager chooses to invite
- SRE team

Duration 1 hour (includes a 15-minute Q&A session)

Preparation Prior to the project briefing, the following must be completed:

By site coordinator

- The completed project profile is given to the SRE team.
- The project briefing content is given to the project manager.

By project manager

- The briefing presenter is chosen and briefing materials are complete.
- The briefing attendees are selected.
- The SRE participants are selected.

Logistics

The site coordinator is responsible for arranging the project briefing with the project manager.

Approach

This briefing will be run by the project manager or a chosen delegate. If, after the presentation, you still do not know the answers to the following questions, ask the presenter:

- ***Who?*** Who are the people the team will be interviewing and where do they fit into the project organization and operations? Who is the customer?
- ***What?*** What is the product this project is making? What are its special features? What makes it a challenge?
- ***Where?*** Where is the work being done? Where will the product be delivered?
- ***When?*** When must the product be delivered to the customer? What are the milestones and contractual dates of the project? Where is the project in its schedule right now?
- ***How?*** How is the project team developing the product? What processes is it following?
- ***What is the project's "picture of success?"*** This should be stated succinctly in two or three written sentences.

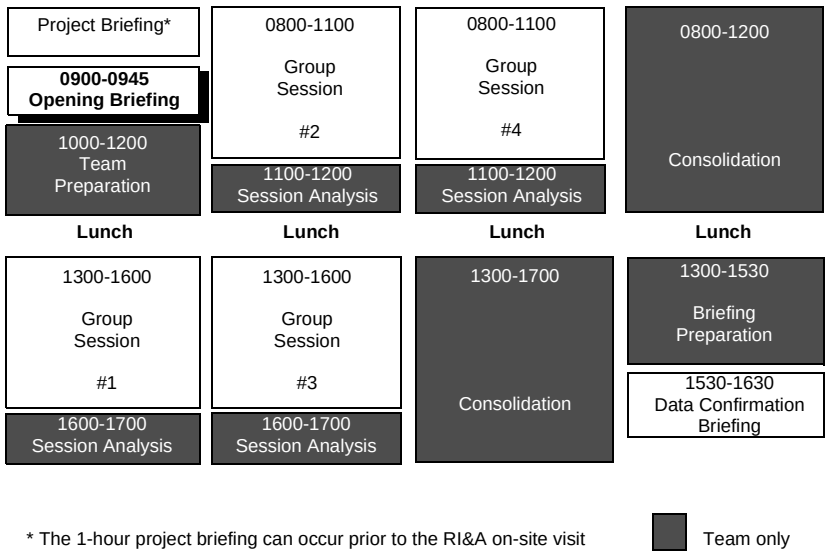
Results

The SRE team has answers to the questions listed above.

Key Considerations

It is likely that the project manager has a "set piece" briefing on hand that is used in various forms to inform outsiders about the project. This usually makes a good starting point for the project briefing. However, if the SRE team needs specific information that will serve as a context for group sessions, make sure that the project manager is asked to give the needed information.

Opening Briefing



Objectives

- to demonstrate management's commitment to the SRE activity
- to set the participants at ease by familiarizing them with the SRE process and its outputs
- to review schedules - where to be, and when
- to answer questions

Who's in the Room?

- project manager (required)
- all individuals who will participate (strongly recommended)
- other project personnel (recommended, but optional)
- SRE team.

Duration

45 minutes

Preparation

Prior to the opening briefing, the following must be completed:

- All on-site logistics arrangements have been made by the site coordinator.
- Briefing materials have been completed.

Approach

The project manager will introduce SRE team members and demonstrate his or her management commitment to the activity. The SRE team leader will then take over and deliver the briefing, which should take 30 minutes. This allows 15 minutes for questions.

The opening briefing should cover the following:

- the benefits of conducting an SRE
- the products of an SRE
- the SRE process
- what to expect from an SRE
- why an SRE is used
- the schedule

After the briefing, take any questions from the audience.

Results

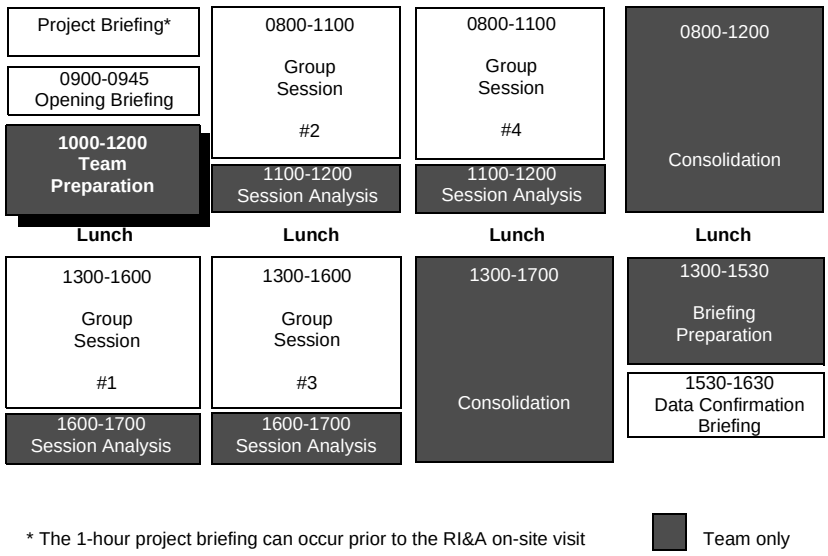
Participants understand the following

- that management is committed
- what to expect during the SRE process
- where to be and when

Key Considerations

- It is key that the project manager visibly commits to the process and introduces the SRE team. Showing commitment encourages project personnel to participate fully in the process. If the management isn't committed, why should the project personnel participate?
- Allow ample time for questions from the audience. The purpose is to set participants in the process at ease about what to expect and what is expected of them.

Team Preparation



Objectives to finalize any last minute preparations for the RI&A phase

Who's in the Room? SRE team

Duration 3 hours maximum

Preparation All on-site logistical arrangements must be completed by the site coordinator before the team preparation begins.

Results All team members know exactly what will happen in each activity and what their roles and responsibilities are.

Points to Remember This is the only time the team will have an extended period of time to "sit back" and take a look at what is going to happen. The following three

days of on-site activity are fit into a tight schedule. Use the time to make sure that team members are in synch with each other. A prepared team is a more effective team.

Logistics If possible, take a look at the rooms assigned for each activity. Knowing the layout of the rooms ahead of time will minimize the set-up time later.

Forms to Be Used The project profile shown on page 11 is used.

Project Profile

1. What are the normal work hours of the project (e.g., 8:00-5:00)? _____

2. What is your project's contractual role?

   ❒ Prime ❒ Subcontractor

   ❒ Integrator Other: _____

3 What are the start and delivery dates for your project?

   Start: _____ Delivery: _____

4 What phases does the contract life cycle cover?

   Demonstration and validation ❒yes❒no

   Full-scale development ❒yes❒no

   Maintenance ❒yes❒no

   Other: _____

5 What is the current phase of your project?

   _____

6 Specifically, are you *in* or *past* the implementation phase of your project?

   ❒in ❒past

7 Has your company implemented other systems of this application type?

   ❒yes ❒no

8 Has your company built other systems of this size?

   ❒yes ❒no

8 How big is the software portion of your project?

Number of CSCIs	LOC	Number of CSCs
_____	_____	_____

10 Are there any requirements that require unprecedented or state-of-the-art technology to implement?

Technologies	❒yes	❒no
Methods	❒yes	❒no
Languages	❒yes	❒no

11 Are you using any reused or reengineered software?

❒yes ❒no

12 Are you using any COTS software?

❒yes ❒no

13 Is any developmental hardware being used?

❒yes ❒no

14 Are you doing any prototyping?

❒yes ❒no

15 Are there distributed development sites?

❒yes ❒no

16 Do you have any associate contractors?

❒yes ❒no

17 Do you have any subcontractors?

❒yes ❒no

17 Are any security requirements allocated to the software?

❒yes ❒no

18 Are any safety requirements allocated to the software?

❒yes ❒no

18 Are there multiple installation sites?

❒yes ❒no

Using the Project Profile to Delete Questions from the Taxonomy-Based Questionnaire

Description

When the SRE team receives the filled-out project profile from the project, the information in the profile can be used to eliminate some questions that would otherwise be asked in the group session interview.

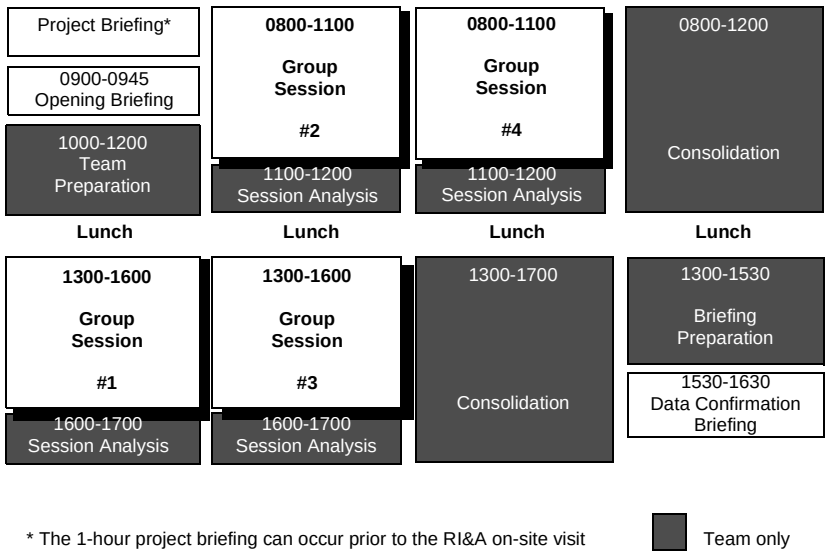
Procedure

The following table defines which answers to the profile's questions can permit questions in the Taxonomy-Based Questionnaire (TBQ) to be skipped. No other answers in the profile have any effect on the TBQ—they only provide general data that may be useful to the SRE team to know before the RI&A phase.

Caution: **Make sure that crossed-out questions on the interviewers' copies remain legible.** In the course of the interview, the team may learn that one or more of the questions was incorrectly eliminated, and legibility will permit their immediate reintroduction.

For this profile question...	if the answer is…	cross out these TBQ questions.
2. What is your project's contractual role?	NOT subcontrac-tor	184 - 187
6. Specifically, are you *in* or *past* the implementation phase of your project?	No	76
11. Are you using any reused or reengineered software?	No	28
12. Are you using any COTS software?	No	29 - 30 55
13. Is any developmental hardware being used?	No	43 - 44
14. Are you doing any prototyping?	No	71.a.1 - 71.a.1a.3
15. Are there distributed development sites?	No	83
16. Do you have any associate contractors?	No	175 - 177
17. Do you have any subcontractors?	No	178 - 183
18. Are any security requirements allocated to the software?	No	68-70
19. Are any safety requirements allocated to the software?	No	66-67
20. Are there multiple installation sites?	No	132

Group Sessions



Objectives

- to elicit risks from project members in an efficient, repeatable, and non-judgemental way
- to facilitate the individual analysis of risks from participants

Who's in the Room?

- peer group of one to five participants
- SRE team

Duration

3 hours

Preparation

The following things must be completed prior to conducting a group session:

- Team roles must be assigned for the interviewer, risk recorder, and session recorder (may rotate for each group session).
- The group session script must be filled out by the interviewer.

- The medium for capturing risk statements must be selected (e.g., flip chart and marker).
- The blank evaluation form must be ready to be filled in.

Approach

The interviewer conducts the interview; the risk recorder captures the risk statements; and the session recorder captures the context of each risk. The three-hour group session should break down as follows:

Opening the Group Session: 5 minutes (see the Group Session Introduction Script on page 20)

- Welcome participants.
- Introduce the members of the team.
- Explain confidentiality and non-attribution.
- Describe the group session.
- Explain the interview process.
- Describe how to construct risk statements.
- Explain the focus during identification.
- Explain how the interview will end.
- Announce the starting point in the TBQ.

Identifying Risks in the Interview: 2 hours and 25 minutes

- Read questions verbatim from the TBQ.
- Use the interview protocol to probe for risks (cues and follow-up questions).
- Determine whether participants want to identify a risk. If they do, capture their risk statements.
- Ask the next question in the TBQ.
- Repeat until you finish the TBQ, or there are 10 minutes remaining in the allotted interview time.
- If you do not finish the TBQ, hand out a copy of the TBQ structure and ask the question: “Are there any concerns or issues you would like to raise beyond those already listed?”
- Capture any new risk statements.

Participant Break: 5 minutes

- Declare a 5 minute break for participants.
- Add the new risk statements to the evaluation form.
- Reproduce copies of the evaluation form.

Analyzing the Risks: 20 minutes (see Group Session Analysis Script on page 55)

- Distribute an evaluation form, scoring matrix, and impact definitions handout to each participant.
- Explain how to evaluate the probability and impact for each risk.
- Explain how to select the “top 5” risks to the project.
- Ask participants to hand in the forms when finished.

Closing the Group Session: 5 minutes (see Closing Script on page 56)

- Thank participants.
- Remind participants about confidentiality and non-attribution.
- Remind participants about the data confirmation briefing day, time, and location.

The session recorder(s) are responsible for reproducing and distributing copies of the context notes to SRE team members. If there were two or more session recorders, these notes should be merged to create a single version. Every attempt should be made to produce notes for the day’s sessions before the end of that day. It is strongly recommended that context note capture (by both the session recorder and other team members) be done with a laptop computer. This will allow the notes from all sources to be reconciled and combined quickly, and make it possible for each SRE team member to have a *legible* copy of the context when consolidation begins.

Team members are responsible for reading the context notes of all sessions before the start of the reconcile scoring activity.

Results

- a list of project risk statements
- context notes for each risk statement

- a completed risk evaluation form for each participant

Key Considerations

- Keep in mind the interview principle: the interviewer must try to balance the following
  - good risk statement quality (condition; consequence)
  - the number of risks identified
  - covering the TBQ
- Keep in mind the individual voice principle and consensus: Any participant in a group may identify a risk. Consensus of the other participants is needed only in the wording of the risk, not in whether it is a risk.
- Capturing the first risk statement sets the tone for the interview. It is important that the participants and not the team identify risks. Use the words of the participants in capturing the risk statements. Ask them how they would phrase the risk and encourage them to modify a statement if it does not reflect what they said.
- Remember, the SRE team works together to identify risks. The rapport between the interviewer and risk recorder is especially important since they interact directly with the participants.
- There should not be any discussion among participants during the analysis. Each participant should evaluate the attributes and top five risks to the project individually and independently.

Logistics

- It is important that the participants be able to see what the risk recorder is writing.
- Identify the number of each question asked in the TBQ. It helps the session recorder to keep track of where the risk context fits.
- If possible, keep all risk statements visible to the participants. This allows them to review what they have already identified.
- If possible, add the risk statements to the evaluation form as they are identified. It will save you time at the end of the interview. This can be accomplished easily if there is an extra team member. If not, the risk recorder may be able to transcribe the risk onto the evaluation form during the interview.
- Access to copy machines, computers, and printers will keep the activity running smoothly.

Scripts and Forms

The following pages provide

- a script for introducing the group session
- the complete TBQ
- instructions for using several questioning techniques as aids to closing the interview and assuring that all the software risk taxonomy elements are covered during the interview
- a copy of the taxonomy outline
- a blank evaluation form
- an example of a filled-in session recorder's notes page
- the generic risk scoring matrix
- the generic levels of risk impact table
- a script for closing the group session

NOTE: The last two items are "generic" because they may have been superseded by project-specific versions during SRE contracting. If they *have* been superseded, the SRE team leader will provide you with the correct versions.

Blank session recorder notes pages appear at the back of this TMNB beginning on page 145.

Group Session Introduction Script

Welcome

- Thank you for being here.
- My name is _____. I'm the interviewer for this session.
- Do you all have name cards?

Introduction

- I'd like to introduce the Software Risk Evaluation team.

_____ is the risk recorder.

_____ is the session recorder.

_____ is the process observer .

Other team members include

_____

_____

_____.

- Now I'd like each of you to introduce yourself and briefly describe your function on the project.

Confidentiality

- Remember that this SRE team and your project have agreed that these sessions will remain confidential.
- We will not attribute any remark to any individual or to this group—even among ourselves after the SRE process is completed. We ask that you follow the same guidelines among yourselves.

Session Description

- This group session consists of a two-and-a-half hour structured interview for risk identification. During this time you will help us write risk statements that relate to your project.
- This will be followed by an analysis phase, in which you will individually analyze attributes of the risk statements you have helped to write.
- Finally, you will individually select and rank five of those risk statements as the "most important to the project."
- The whole group session will take three hours.

Interview Process

- The purpose of this interview is to ask you to identify project issues that reflect your perspective on concerns, uncertainties, or risks that you feel the project is facing.
- We do this using a questionnaire that is structured according to the risk taxonomy.
- I will be asking the taxonomy questions during this session; however, my colleagues may ask follow-up questions at any time.

Constructing Risk Statements

- When you identify an issue and our discussion leads to the construction of a **risk statement**, it will be recorded on a flipchart for you to edit and confirm. It is important that the words accurately reflect what you intended.
- The general format of a risk statement is a phrase describing a **condition** that exists today in the project, followed by a phrase describing at least one possible future **consequence** of that condition. A simple (and non-technical) example might be, "There is water on the hall floor; someone could slip on it and fall."
- Remember that the identification of risks does not require consensus; any one of you may bring up an issue and help us refine it into a risk statement.
- However, it is important that you all agree on the meaning of the risk statement, as reflected in the wording, whether or not you individually agree that it is valid.

Identification Focus

- I want to remind you of your project's "Picture of Success," which is _____. The focus of our discussion should be on things that may jeopardize your reaching that goal.
- We encourage the free flow of responses during the interview, so don't restrict yourself by addressing only the question that was asked. Think of the questions as prompts to stimulate your ideas in the spirit of brainstorming.
- Not every question is expected to lead to the creation of a risk statement. If you don't think there is a concern in an area, just tell me and I'll move on. If you think there ***is*** a concern, bring it up and we'll explore it further.
- It's my job to keep the interview focused on identifying issues. To keep to our deadline, I may interrupt to redirect lengthy discussions or conjecture about solutions.

- And if I get caught up in your discussion, one of my colleagues will suggest that I move on.

Closing the Interview

- We may not complete the questionnaire in the time allotted.
- If we are 10 minutes away from the end of the interview session, and still have not completed the questionnaire, we will interrupt the taxonomy-based interview process and ask if there are any concerns or issues that you would like to raise beyond those already listed.
- Then we will move on to the analysis phase of the session that I mentioned earlier.

Taxonomy Questionnaire

- Do you have any questions before we start?
- We will start with questions from the _____ class of the taxonomy. The first question is from the _____ element and deals with _____ (attribute).

Taxonomy-Based Questionnaire

This is a reprint of Appendix B, Taxonomy-Based Questionnaire, taken from the following technical report: *Taxonomy Based Risk Identification* (CMU/SEI-93-TR-6).

A Product Engineering

A.1 Requirements

A.1-a. Stability

Are requirements changing even as the product is being produced?

[1] Are the requirements stable?

(No) (1.a) What is the effect on the system?

- *Quality*
- *Functionality*
- *Schedule*
- *Integration*
- *Design*
- *Testing*

[2] Are the external interfaces changing?

A.1-b. Completeness

Are requirements missing or incompletely specified?

[3] Are there any TBDs in the specifications?

[4] Are there requirements you know should be in the specification but aren't?

(Yes) (4.a) Will you be able to get these requirements into the system?

[5] Does the customer have unwritten requirements/expectations?

(Yes) (5.a) Is there a way to capture these requirements?

[6] Are the external interfaces completely defined?

A.1-c. Clarity

Are requirements unclear or in need of interpretation?

[7] Are you able to understand the requirements as written?

(No) (7.a) Are the ambiguities being resolved satisfactorily?

(Yes) (7.b) There are no ambiguities or problems of interpretation?

A.1-d. Validity

Will the requirements lead to the product the customer has in mind?

[8] Are there any requirements that may not specify what the customer really wants?

(Yes) (8.a) How are you resolving this?

[9] Do you and the customer understand the same thing by the requirements?

(Yes) (9.a) Is there a process by which to determine this?

[10] How do you validate the requirements?

- Prototyping
- Analysis
- Simulations

A.1-e. Feasibility

Are requirements infeasible from an analytical point of view?

[11] Are there any requirements that are technically difficult to implement?

(Yes) (11.a) What are they?

(Yes) (11.b) Why are they difficult to implement?

(No) (11.c) Were feasibility studies done for these requirements?

(Yes) (11.c.1) How confident are you of the assumptions made in the studies?

A.1-f. Precedent

Do requirements specify something never done before, or that your company has not done before?

[12] Are there any state-of-the-art requirements?

- Technologies
- Methods
- Languages
- Hardware

(No) (12.a) Are any of these new to you?

(Yes) (12.b) Does the program have sufficient knowledge in these areas?

(No) (12.b.1) Is there a plan for acquiring knowledge in these areas?

A.1-g. Scale

Do requirements specify a product larger, more complex, or requiring a larger organization than in the experience of the company?

[13] Is the system size and complexity a concern?

(No) (13.a) Have you done something of this size and complexity before?

[14] Does the size require a larger organization than usual for your company?

A.2 Design

A.2-a. Functionality

Are there any potential problems in meeting functionality requirements?

[15] Are there any specified algorithms that may not satisfy the requirements?

(No) (15.a) Are any of the algorithms or designs marginal with respect to meeting requirements?

[16] How do you determine the feasibility of algorithms and designs?

- Prototyping
- Modeling
- Analysis
- Simulation

A.2-b. **Difficulty**

Will the design and/or implementation be difficult to achieve?

[17] Does any of the design depend on unrealistic or optimistic assumptions?

[18] Are there any requirements or functions that are difficult to design?

(No) (18.a) Do you have solutions for all the requirements?

(Yes) (18.b) What are the requirements?

- Why are they difficult?

A.2-c. **Interfaces**

Are the internal interfaces (hardware and software) well defined and controlled?

[19] Are the internal interfaces well defined?

- Software-to-software
- Software-to-hardware

[20] Is there a process for defining internal interfaces?

(Yes) (20.a) Is there a change control process for internal interfaces?

[21] Is hardware being developed in parallel with software?

(Yes) (21.a) Are the hardware specifications changing?

(Yes) (21.b) Have all the interfaces to software been defined?

(Yes) (21.c) Will there be engineering design models that can be used to test the software?

A.2-d. **Performance**

Are there stringent response time or throughput requirements?

[22] Are there any problems with performance?

- Throughput
- Scheduling asynchronous real-time events
- Real-time response
- Recovery timelines
- Response time
- Database response, contention, or access

[23] Has a performance analysis been done?

(Yes) (23.a) What is your level of confidence in the performance analysis?

(Yes) (23.b) Do you have a model to track performance through design and implementation?

A.2-e. **Testability**
Is the product difficult or impossible to test?

[24] Is the software going to be easy to test?

[25] Does the design include features to aid testing?

[26] Do the testers get involved in analyzing requirements?

A.2-f. **Hardware Constraints**
Are there tight constraints on the target hardware?

[27] Does the hardware limit your ability to meet any requirements?

- Architecture
- Memory capacity
- Throughput
- Real-time response
- Response time
- Recovery timelines
- Database performance
- Functionality
- Reliability
- Availability

A.2-g. **Non-Developmental Software**
Are there problems with software used in the program but not developed by the program?

If reused or reengineered software exists

[28] Are you reusing or re-engineering software not developed on the program?

(Yes) (28.a) Do you foresee any problems?

- Documentation

- Performance
- Functionality
- Timely delivery
- Customization

If COTS software is being used

[29] Are there any problems with using COTS (commercial off-the-shelf) software?
- Insufficient documentation to determine interfaces, size, or performance
- Poor performance
- Requires a large share of memory or database storage
- Difficult to interface with application software
- Not thoroughly tested
- Not bug free
- Not maintained adequately
- Slow vendor response

[30] Do you foresee any problem with integrating COTS software updates or revisions?

A.3 Code and Unit Test

A.3-a. Feasibility

Is the implementation of the design difficult or impossible?

[31] Are any parts of the product implementation not completely defined by the design specification?

[32] Are the selected algorithms and designs easy to implement?

A.3-b. Testing

Are the specified level and time for unit testing adequate?

[33] Do you begin unit testing before you verify code with respect to the design?

[34] Has sufficient unit testing been specified?

[35] Is there sufficient time to perform all the unit testing you think should be done?

[36] Will compromises be made regarding unit testing if there are schedule problems?

A.3-c. **Coding/Implementation**
Are there any problems with coding and implementation?

[37] Are the design specifications in sufficient detail to write the code?

[38] Is the design changing while coding is being done?

[39] Are there system constraints that make the code difficult to write?
- Timing
- Memory
- External storage

[40] Is the language suitable for producing the software on this program?

[41] Are there multiple languages used on the program?

(Yes) (41.a) Is there interface compatibility between the code produced by the different compilers?

[42] Is the development computer the same as the target computer?

(No) (42.a) Are there compiler differences between the two?

If developmental hardware is being used

[43] Are the hardware specifications adequate to code the software?

[44] Are the hardware specifications changing while the code is being written?

A.4 Integration and Test

A.4-a. Environment

Is the integration and test environment adequate?

[45] Will there be sufficient hardware to do adequate integration and testing?

[46] Is there any problem with developing realistic scenarios and test data to demonstrate any requirements?

- Specified data traffic
- Real-time response
- Asynchronous event handling
- Multi-user interaction

[47] Are you able to verify performance in your facility?

[48] Does hardware and software instrumentation facilitate testing?

(Yes) (48.a) Is it sufficient for all testing?

A.4-b. Product

Is the interface definition inadequate, facilities inadequate, time insufficient?

[49] Will the target hardware be available when needed?

[50] Have acceptance criteria been agreed to for all requirements?

(Yes) (50.a) Is there a formal agreement?

[51] Are the external interfaces defined, documented, and baselined?

[52] Are there any requirements that will be difficult to test?

[53] Has sufficient product integration been specified?

[54] Has adequate time been allocated for product integration and test?

If COTS

[55] Will vendor data be accepted in verification of requirements allocated to COTS products?

(Yes) (55.a) Is the contract clear on that?

A.4-c. System

System integration uncoordinated, poor interface definition, or inadequate facilities?

[56] Has sufficient system integration been specified?

[57] Has adequate time been allocated for system integration and test?

[58] Are all contractors part of the integration team?

[59] Will the product be integrated into an existing system?

(Yes) (59.a) Is there a parallel cutover period with the existing system?

(No) (59.a.1) How will you guarantee the product will work correctly when integrated?

[60] Will system integration occur on customer site?

A.5 Engineering Specialties

A.5-a. Maintainability

Will the implementation be difficult to understand or maintain?

[61] Does the architecture, design, or code create any maintenance difficulties?

[62] Are the maintenance people involved early in the design?

[63] Is the product documentation adequate for maintenance by an outside organization?

A.5-b. Reliability

Are the reliability or availability requirements difficult to meet?

[64] Are reliability requirements allocated to the software?

[65] Are availability requirements allocated to the software?

(Yes) (65.a) Are recovery timelines any problem?

A.5-c. **Safety**

Are the safety requirements infeasible and not demonstrable?

[66] Are safety requirements allocated to the software?

(Yes) (66.a) Do you see any difficulty in meeting the safety requirements?

[67] Will it be difficult to verify satisfaction of safety requirements?

A.5-d. **Security**

Are the security requirements more stringent than the current state of the practice or program experience?

[68] Are there unprecedented or state-of-the-art security requirements?

[69] Is it an Orange Book system?

[70] Have you implemented this level of security before?

A.5-e. **Human Factors**

Will the system will be difficult to use because of poor human interface definition?

[71] Do you see any difficulty in meeting the Human Factors requirements?

(No) (71.a.0) How are you ensuring that you will meet the human interface requirements?

If prototyping

(No) (71.a.1) Is it a throw-away prototype?

(No) (71.a.1a) Are you doing evolutionary development?

(Yes) (71.a.1a.1) Are you experienced in this type of development?

(Yes) (71.a.1a.2) Are interim versions deliverable?

(Yes) (71.a.1a.3) Does this complicate change control?

A.5-f. **Specifications**

Is the documentation adequate to design, implement, and test the system?

[72] Is the software requirements specification adequate to design the system?

[73] Are the hardware specifications adequate to design and implement the software?

[74] Are the external interface requirements well specified?

[75] Are the test specifications adequate to fully test the system?

If in or past implementation phase

[76] Are the design specifications adequate to implement the system?

- Internal interfaces

B. Development Environment

B.1 Development Process

B.1-a. Formality

Will the implementation be difficult to understand or maintain?

[77] Is there more than one development model being used?

- Spiral
- Waterfall
- Incremental

(Yes) (77.a) Is coordination between them a problem?

[78] Are there formal, controlled plans for all development activities?

- Requirements analysis
- Design
- Code
- Integration and test
- Installation
- Quality assurance
- Configuration management

(Yes) (78.a) Do the plans specify the process well?

(Yes) (78.b) Are developers familiar with the plans?

B.1-b. Suitability

Is the process suited to the development model, e.g., spiral, prototyping?

[79] Is the development process adequate for this product?

[80] Is the development process supported by a compatible set of procedures, methods, and tools?

B.1-c. **Process Control**

Is the software development process enforced, monitored, and controlled using metrics? Are distributed development sites coordinated?

[81] Does everyone follow the development process?

(Yes) (81.a) How is this insured?

[82] Can you measure whether the development process is meeting your productivity and quality goals?

If there are distributed development sites

[83] Is there adequate coordination among distributed development sites?

B.1-d. **Familiarity**

Are the project members experienced in use of the process? Is the process understood by all staff members?

[84] Are people comfortable with the development process?

B.1-e. **Product Control**

Are there mechanisms for controlling changes in the product?

[85] Is there a requirements traceability mechanism that tracks requirements from the source specification through test cases?

[86] Is the traceability mechanism used in evaluating requirement change impact analyses?

[87] Is there a formal change control process?

(Yes) (87.a) Does it cover all changes to baselined requirements, design, code, and documentation?

[88] Are changes at any level mapped up to the system level and down through the test level?

[89] Is there adequate analysis when new requirements are added to the system?

[90] Do you have a way to track interfaces?

[91] Are the test plans and procedures updated as part of the change process?

B.2 Development System

B.2-a. Capacity

Is there sufficient work station processing power, memory, or storage capacity?

[92] Are there enough workstations and processing capacity for all staff?

[93] Is there sufficient capacity for overlapping phases, such as coding, integration and test?

B.2-b. Suitability

Does the development system support all phases, activities, and functions?

[94] Does the development system support all aspects of the program?

- Requirements analysis
- Performance analysis
- Design
- Coding
- Test
- Documentation
- Configuration management
- Management tracking
- Requirements traceability

B.2-c. Usability

How easy is the development system to use?

[95] Do people find the development system easy to use?

[96] Is there good documentation of the development system?

B.2-d. **Familiarity**

Is there little prior company or project member experience with the development system?

[97] Have people used these tools and methods before?

B.2-e. **Reliability**

Does the system suffer from software bugs, down-time, insufficient built-in back-up?

[98] Is the system considered reliable?

- Compiler
- Development tools
- Hardware

B.2-f. **System Support**

Is there timely expert or vendor support for the system?

[99] Are the people trained in use of the development tools?

[100] Do you have access to experts in use of the system?

[101] Do the vendors respond to problems rapidly?

B.2-g. **Deliverability**

Are the definition and acceptance requirements defined for delivering the development system to the customer not budgeted? HINT: If the participants are confused about this, it is probably not an issue from a risk perspective.

[102] Are you delivering the development system to the customer?

(Yes) (102.a) Have adequate budget, schedule, and resources been allocated for this deliverable?

B.3 Management Process

B.3-a. Planning

Is the planning timely, technical leads included, contingency planning done?

[103] Is the program managed according to the plan?

(Yes) (103.a) Do people routinely get pulled away to fight fires?

[104] Is re-planning done when disruptions occur?

[105] Are people at all levels included in planning their own work?

[106] Are there contingency plans for known risks?

(Yes) (106.a) How do you determine when to activate the contingencies?

[107] Are long-term issues being adequately addressed?

B.3-b. Project Organization

Are the roles and reporting relationships clear?

[108] Is the program organization effective?

[109] Do people understand their own and others' roles in the program?

[110] Do people know who has authority for what?

B.3-c. Management Experience

Are the managers experienced in software development, software management, the application domain, the development process, or on large programs?

[111] Does the program have experienced managers?

- Software management
- Hands-on software development
- With this development process
- In the application domain
- Program size or complexity

B.3-d. Program Interfaces

Is there poor interface with customer, other contractors, senior and/or peer managers?

[112] Does management communicate problems up and down the line?

[113] Are conflicts with the customer documented and resolved in a timely manner?

[114] Does management involve appropriate program members in meetings with the customer?

- Technical leaders
- Developers
- Analysts

[115] Does management work to ensure that all customer factions are represented in decisions regarding functionality and operation?

[116] Is it good politics to present an optimistic picture to the customer or senior management?

B.4 Management Methods

B.4-a. Monitoring

Are management metrics defined and development progress tracked?

[117] Are there periodic structured status reports?

(Yes) (117.a) Do people get a response to their status reports?

[118] Does appropriate information get reported to the right organizational levels?

[119] Do you track progress versus plan?

(Yes) (119.a) Does management have a clear picture of what is going on?

B.4-b. Personnel Management

Are project personnel trained and used appropriately?

[120] Do people get trained in skills required for this program?

(Yes) (120.a) Is this part of the program plan?

[121] Do people get assigned to the program who do not match the experience profile for your work area?

[122] Is it easy for program members to get management action?

[123] Are program members at all levels aware of their status versus plan?

[124] Do people feel it's important to keep to the plan?

[125] Does management consult with people before making decisions that affect their work?

[126] Does program management involve appropriate program members in meetings with the customer?

- Technical leaders
- Developers
- Analysts

B.4-c. Quality Assurance

Are there adequate procedures and resources to assure product quality?

[127] Is the software quality assurance function adequately staffed on this program?

[128] Do you have defined mechanisms for assuring quality?

(Yes) (128.a) Do all areas and phases have quality procedures?

(Yes) (128.b) Are people used to working with these procedures?

B.4-d. Configuration Management

Are the change procedures or version control, including installation site(s), adequate?

[129] Do you have an adequate configuration management system?

[130] Is the configuration management function adequately staffed?

[131] Is coordination required with an installed system?

(Yes) (131.a) Is there adequate configuration management of the installed system?

(Yes) (131.b) Does the configuration management system synchronize your work with site changes?

[132] Are you installing in multiple sites?

(Yes) (132.a) Does the configuration management system provide for multiple sites?

B.5 Work Environment

B.5-a. Quality Attitude

Is there a lack of orientation toward quality work?

[133] Are all staff levels oriented toward quality procedures?

[134] Does schedule get in the way of quality?

B.5-b. Cooperation

Is there a lack of team spirit? Does conflict resolution require management intervention?

[135] Do people work cooperatively across functional boundaries?

[136] Do people work effectively toward common goals?

[137] Is management intervention sometimes required to get people working together?

B.5-c. Communication

Is there poor awareness of mission or goals, poor communication of technical information among peers and managers?

[138] Is there good communication among the members of the program?

- Managers
- Technical leaders
- Developers
- Testers
- Configuration management

- Quality assurance

[139] Are the managers receptive to communication from program staff?

(Yes) (139.a) Do you feel free to ask your managers for help?

(Yes) (139.b) Are members of the program able to raise risks without having a solution in hand?

[140] Do the program members get timely notification of events that may affect their work?

(Yes) (140.a) Is this formal or informal?

B.5-d. **Morale**

Is there a non-productive, non-creative atmosphere? Do people feel that there is no recognition or reward for superior work?

[141] How is morale on the program?

(No) (141.a) What is the main contributing factor to low morale?

[142] Is there any problem keeping the people you need?

C. Program Constraints

C.1 Resources

C.1-a. Schedule

Is the schedule inadequate or unstable?

[143] Has the schedule been stable?

[144] Is the schedule realistic?

(Yes) (144.a) Is the estimation method based on historical data?

(Yes) (144.b) Has the method worked well in the past?

[145] Is there anything for which adequate schedule was not planned?

- Analysis and studies
- QA
- Training
- Maintenance courses and training
- Capital equipment
- Deliverable development system

[146] Are there external dependencies which are likely to impact the schedule?

C.1-b. Staff

Is the staff inexperienced, lacking domain knowledge, lacking skills, or understaffed?

[147] Are there any areas in which the required technical skills are lacking?

- Software engineering and requirements analysis method
- Algorithm expertise
- Design and design methods
- Programming languages
- Integration and test methods
- Reliability
- Maintainability
- Availability
- Human factors
- Configuration management
- Quality assurance

- Target environment
- Level of security
- COTS
- Reuse software
- Operating system
- Database
- Application domain
- Performance analysis
- Time-critical applications

[148] Do you have adequate personnel to staff the program?

[149] Is the staffing stable?

[150] Do you have access to the right people when you need them?

[151] Have the program members implemented systems of this type?

[152] Is the program reliant on a few key people?

[153] Is there any problem with getting cleared people?

C.1-c. **Budget**
Is the funding insufficient or unstable?

[154] Is the budget stable?

[155] Is the budget based on a realistic estimate?

(Yes) (155.a) Is the estimation method based on historical data?

(Yes) (155.b) Has the method worked well in the past?

[156] Have features or functions been deleted as part of a design-to-cost effort?

[157] Is there anything for which adequate budget was not allocated?

- Analysis and studies
- QA
- Training
- Maintenance courses
- Capital equipment
- Deliverable development system

[158] Do budget changes accompany requirement changes?

(Yes) (158.a) Is this a standard part of the change control process?

C.1-d. **Facilities**

Are the facilities adequate for building and delivering the product?

[159] Are the development facilities adequate?

[160] Is the integration environment adequate?

C.2 **Contract**

C.2-a. **Type of Contract**

Is the contract type a source of risk to the program?

[161] What type of contract do you have? (Cost plus award fee, fixed price,....)

(161a) Does this present any problems?

[162] Is the contract burdensome in any aspect of the program?

- SOW (Statement of Work)
- Specifications
- DIDs (Data Item Descriptions)
- Contract parts
- Excessive customer involvement

[163] Is the required documentation burdensome?

- Excessive amount
- Picky customer
- Long approval cycle

C.2-b. **Restrictions**

Does the contract cause any restrictions?

[164] Are there problems with data rights?

- COTS software
- Developmental software
- Non-developmental items

C.2-c. Dependencies

Does the program have any dependencies on outside products or services?

[165] Are there dependencies on external products or services that may affect the product, budget, or schedule?

- Associate contractors
- Prime contractor
- Subcontractors
- Vendors or suppliers
- Customer furnished equipment or software

C.3 Program Interfaces

C.3-a. Customer

Are there any customer problems such as: lengthy document-approval cycle, poor communication, and inadequate domain expertise?

[166] Is the customer approval cycle timely?

- Documentation
- Program reviews
- Formal reviews

[167] Do you ever proceed before receiving customer approval?

[168] Does the customer understand the technical aspects of the system?

[169] Does the customer understand software?

[170] Does the customer interfere with process or people?

[171] Does management work with the customer to reach mutually agreeable decisions in a timely manner?

- Requirements understanding
- Test criteria
- Schedule adjustments
- Interfaces

[172] How effective are your mechanisms for reaching agreements with the customer?

- Working groups (contractual?)
- Technical interchange meetings (contractual?)

[173] Are all customer factions involved in reaching agreements?

(Yes) (173.a) Is it a formally defined process?

[174] Does management present a realistic or optimistic picture to the customer?

If there are associate contractors

C.3-b. Associate Contractors

Are there any problems with associate contractors such as inadequately defined or unstable interfaces, poor communication, or lack of cooperation?

[175] Are the external interfaces changing without adequate notification, coordination, or formal change procedures?

[176] Is there an adequate transition plan?

(Yes) (176.a) Is it supported by all contractors and site personnel?

[177] Is there any problem with getting schedules or interface data from associate contractors?

(No) (177.a) Are they accurate?

If there are subcontractors

C.3-c. Subcontractors

Is the program dependent on subcontractors for any critical areas?

[178] Are there any ambiguities in subcontractor task definitions?

[179] Is the subcontractor reporting and monitoring procedure different from the program's reporting requirements?

[180] Is subcontractor administration and technical management done by a separate organization?

[181] Are you highly dependent on subcontractor expertise in any areas?

[182] Is subcontractor knowledge being transferred to the company?

[183] Is there any problem with getting schedules or interface data from subcontractors?

If program is a subcontract

C.3-d. **Prime Contractor**

Is the program facing difficulties with its Prime contractor?

[184] Are your task definitions from the Prime ambiguous?

[185] Do you interface with two separate prime organizations for administration and technical management?

[186] Are you highly dependent on the Prime for expertise in any areas?

[187] Is there any problem with getting schedules or interface data from the Prime?

C.3-e. **Corporate Management**

Is there a lack of support or micro management from upper management?

[188] Does program management communicate problems to senior management?

(Yes) (188.a) Does this seem to be effective?

[189] Does corporate management give you timely support in solving your problems?

[190] Does corporate management tend to micro-manage?

[191] Does management present a realistic or optimistic picture to senior management?

C.3-f. **Vendors**

Are vendors responsive to programs needs?

[192] Are you relying on vendors for deliveries of critical components?

- Compilers
- Hardware
- COTS

C.3-g. **Politics**

Are politics causing a problem for the program?

[193] Are politics affecting the program?

- Company
- Customer
- Associate contractors
- Subcontractors

[194] Are politics affecting technical decisions?

Ending the Interview—Directions and Script

Objective

The interviewer has to decide on the fly whether the interview is covering the taxonomy well. If only a few classes and elements have been covered when there are only 15 or so minutes left in the interview, it is appropriate to shift the level of inquiry from the ***attribute*** level of the taxonomy (the level at which the TBQ questions are written) to the ***element*** level. Several techniques are available to help ensure coverage of the taxonomy.

Procedure

1. With about 15 minutes remaining (about 2-1/4 hours into the interview), the interviewer will shift to a more unstructured form of questioning. To do this, the interviewer may do one of the following:

   - Use the Short Taxonomy-Based Questionnaire (the "Short TBQ") reproduced on page 53 to shift the level of questioning from the attribute to the element level of the TBQ. Follow the same overall strategy for the order of questioning, and do not ask questions for elements that were already covered completely using the full TBQ.

   OR

   - Place a copy of the taxonomy outline (shown on page 54) in front of the participants and ask them to examine it. Then, go around the table and ask each participant to suggest risk statements for areas that have not yet been covered.

2. After the participants have exhausted their risk issues or the allotted time has been used up, declare a five-minute break. Remind the participants that they ***must*** be back in the room in five minutes and strongly suggest that they not go back to their offices or read email. While they are gone, the computer operator will print the risk evaluation form and make enough copies for everyone in the room.

Script We are just about out of time, so I will stop asking questions from the Taxonomy-Based Questionnaire.

- **Method 1:** I'm now going to switch to a questionnaire that covers the SEI Risk Taxonomy at a higher level, so that we can cover the remaining areas more quickly. I'm going to be asking questions in the _____ Class, beginning with the Element _____. The question is: _____.

OR

- **Method 2:** Here is a copy of the taxonomy outline. Please examine it and then think about any risks that might exist in the areas we have not yet covered. Can you think of any other risks we should capture?.

We are now out of time. Let's take a five-minute break. Please come back after five minutes so that we can keep on schedule. Let me strongly suggest that you do not go back to your desks, go near a telephone, or read email. We'll see you right back here in five minutes. Thank you.

A *Short* Taxonomy-Based Questionnaire

Product (Product Engineering)

Think about risks to the project that may arise from the nature of the product that you are trying to develop...

A.1 Requirements—Are there risks that may arise from requirements being placed on the product? Examples: Stability; Completeness; Clarity; Validity; Feasibility; Precedent; Scale.
A.2 Design—Are there risks that may arise from the design the project has chosen to meet its requirements? Examples: Functionality; Difficulty; Interfaces; Performance; Testability; Hardware Constraints; Non-Developmental Software.
A.3 Code & Unit Test (Manufacturability)—Are there risks that may arise from the way the project is choosing to subdivide the design and construct the pieces? Examples: Feasibility; Testing; Coding/Implementation.
A.4 Integration & Test—Are there risks that may arise from the way the project is choosing to bring the pieces together and prove that they work as a whole? Examples: The HW and SW Support Facilities; integration of the parts of the product; integration with the larger system
A.5 Engineering Specialities—Are there risks that may arise from special attributes of the product, such as Maintainability, Reliability, Safety, Security, Human Factors, etc.?
A.99 (Other)—Are there other risks that may arise from the product itself, but are not covered by the above categories?

Process (Development Environment)

*Think about risks to the project that may arise from the **way** you are going about developing the product...*

B.1 Development Process—Are there risks that may arise from the process the project has chosen to develop the product? Examples: Formality; Suitability; Process Control; Familiarity; Product Control.
B.2 Development System—Are there risks that may arise from the hardware and software tools the project has chosen for controlling and facilitating its development process? Examples: Capacity; Suitability; Usability; Familiarity; Reliability; System Support; Deliverability.
B.3 Management System—Are there risks that may arise from the way project budget or schedule is planned, monitored or controlled, or the project's structure, or its handling of internal and external organization interfaces?
B.4 Management Methods—Are there risks that may arise from the way the development or program personnel are managed, in areas such as Status Monitoring, Personnel Management, Quality Assurance, or Configuration Management?
B.5 Work Environment—Are there risks that may arise from the general environment or the larger organization to which the project belongs, such as Quality Attitude, Cooperation, Communication, or Morale?
B.99 (Other)—Are there other risks that may arise from the way the project is going about its development, but not covered by the above categories?

Constraints (Program Constraints)

Think about risks to the project that may arise from sources outside the project's control...

C.1 Resources—Are there risks that may arise from resources the project needs but that are outside its control to obtain or maintain? Examples: Schedule; Staff; Budget; Facilities.
C.2 Contract—Are there risks that may arise from the [already legally binding] contract? Example areas include the contract's Type, Restrictions, or Dependencies.
C.3 Program Interfaces—Are there risks that may arise from outside interfaces which the project cannot reasonably expect to control? Examples: Customer; Associate Contractors; Subcontractors; Prime Contractor; Corporate Management; Vendors; Politics.
C.99 (Other)—Are there other risks that may arise from factors outside project control, but not covered by the above categories?

Taxonomy of Software Development Risks

A. Product Engineering

1. Requirement
a. Stability
b. Completeness
c. Clarity
d. Validity
e. Feasibility
f. Precedent
g. Scale

1. Design
a. Functionality
b. Difficulty
c. Interfaces
d. Performance
e. Testability
f. Hardware Constraints
g. Non-Developmental Software

1. Code and Unit Test
a. Feasibility
b. Testing
c. Coding/Implementation

1. Integration and Test
a. Environment
b. Product
c. System

1. Engineering Specialties
a. Maintainability
b. Reliability
c. Safety
d. Security
e. Human Factors
f. Specifications

B. Development Environment

1. Development Process
a. Formality
b. Suitability
c. Process Control
d. Familiarity
e. Product Control

1. Development System
a. Capacity
b. Suitability
c. Usability
d. Familiarity
e. Reliability
f. System Support
g. Deliverability

1. Management Process
a. Planning
b. Project Organization
c. Management Experience
d. Program Interfaces

1. Management Methods
a. Monitoring
b. Personnel Management
c. Quality Assurance
d. Configuration Management

1. Work Environment
e. Quality Attitude
f. Cooperation
g. Communication
h. Morale

C. Program Constraints

1. Resources
a. Schedule
b. Staff
c. Budget
d. Facilities

1. Contract
a. Type of Contract
b. Restrictions
c. Dependencies

1. Program Interfaces
a. Customer
b. Associate Contractors
c. Subcontractors
d. Prime Contractor
e. Corporate Management
f. Vendors
g. Politics

Group Session Analysis Directions and Closing Script

Evaluation Forms

- Here is an evaluation form listing all the risk statements you listed during this session.
- The purpose of this analysis is to ask each of you to evaluate the risk statements with respect to two attributes; impact and probability, and then to select the top five most important risks to the program.
- The SRE team is also going to be evaluating the risk statements for impact and probability, and will come up with their own top five risks based on this scoring.

Evaluating the Attributes

- I'm giving you one additional handout to help you in this process. The top half of it is the risk scoring matrix which shows how the scores for impact and probability translate into risk exposure. The bottom half of the handout is the levels of risk impact table, to help us all calibrate what we mean by our impact scores.
- To use the levels of risk impact table, think about the condition that is given in the risk statement and all the consequences that may flow from it; don't limit yourself to just the consequences given in the statement.
- Considering all that could happen as a result of the condition, decide whether you think it is predominantly a risk to ***performance***, ***support*** (supportability or long-term maintainability of the product), ***cost***, or ***schedule***. Once you decide on the impact areas, review the column from that area in the levels of risk impact table. Then, determine whether you think the risk is catastrophic, critical, marginal, or negligible, based on the criteria given. Notice that negligible doesn't mean "zero impact to the program"—it means that it can be handled by built-in margins in the project plan. Too many negligible risks that all come true together can have serious consequences for the program.
- When you have decided on the level of impact, enter its corresponding value (1 to 4) in the Impact column of your risk evaluation form.
- For probability, think in terms of the impact you just decided on.
  - If you think the probability is "somewhere around 50/50," it should be considered "probable," and you should mark a value of "2" in the probability column of your risk evaluation form.

- If you think it's a lot more probable than that, it would be "very likely," and the value to enter is "3."
- If you think it's a lot less probable than 50/50, enter "1" for "improbable."

- Repeat the process for each risk statement.

Choosing the Top Five Risks to the Program

- After evaluating the attributes for each risk statement, select the risk statements which you think point to the greatest threat to the success of the program. Label your top risk statement 1, your next top risk statement 2, and so on.
- It is important to think about how **the risk will affect the program**, rather than just how it will affect you.
- Please hand in the evaluation form when you are finished.
- Are there any questions about how to evaluate the risks?

Closing the Group Session

- Thank you for participating.
- Again, remember that this SRE team and your project have agreed that these sessions will remain confidential. We will keep the conversation inside this room and not attribute any remark to any individual or to this group. We ask that you do the same.
- Finally, don't forget to attend the data confirmation briefing on _____ (day) at _____ (time) in _____ (room).

Risk Evaluation Form

		Team Member's Name ➔		
Risk ID	**Risk Statement**	**Impact**	**Probability**	**Top 5**

Sample Session Recorder Notes

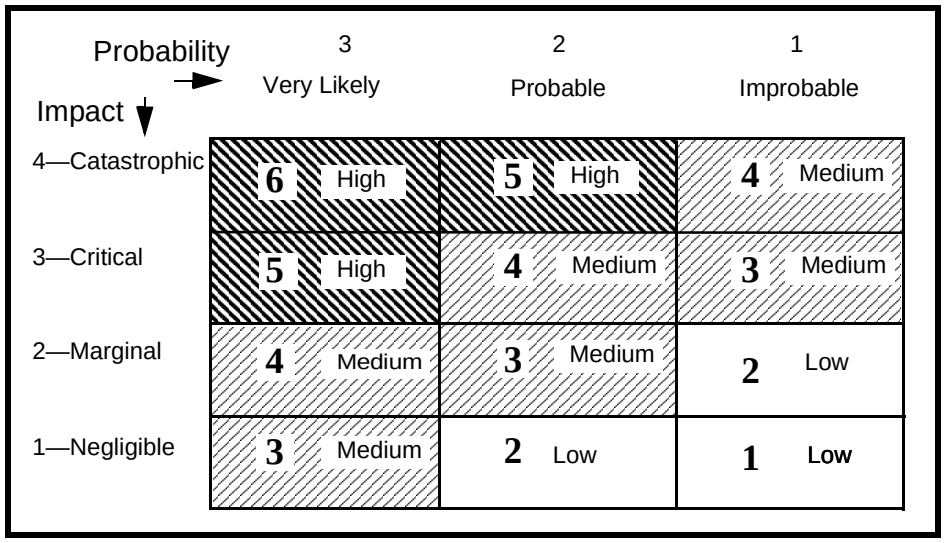
Notes

Q23—Performance analysis: Some "back of the envelope" calculations on problem areas. We may not have targeted all areas. Consequences: We don't know what they'll be; hard to predict where there will be problems (e.g., bottlenecks). There could be a number of consequences. I don't want to put just one—people might think that's all there is. It's a bit premature to nail down one consequence. (Note: Risk Condition only.) R13

Note: This is an example of context captured by the session recorder—an interview participant's comments after being asked question 23 in the TBQ. At the end of—or in the midst of—the discussion, the risk recorder wrote R13 on the flip-chart, indicating risk statement 13. The participant who identified the issue agreed that risk statement 13 was an accurate portrayal of his concern.

Blank session recorder notes pages are provided at the end of this TMNB beginning on page 145.

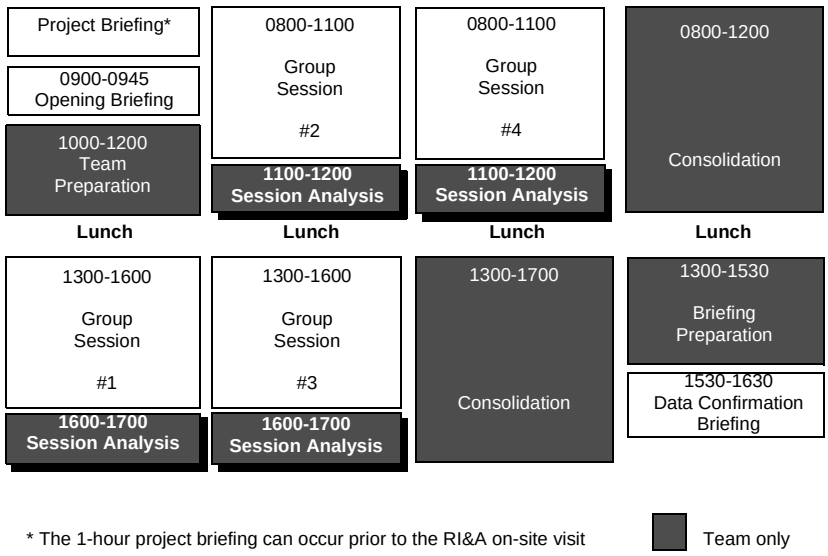
Risk Scoring Matrix



Levels of Risk Impact

Component → / Category ↓	***Performance***	***Support***	***Cost***	***Schedule***
Catastrophic	nonachievement of technical performance	unsupportable software	major budget overrun (>50%)	unachievable IOC
Critical	significant degradation of technical performance	major delays in software modifications	serious budget overrun (~30%)	serious delay in IOC (>30% late)
Marginal	some reduction in technical performance	minor delays in software modifications	budget overrun (~10%)	delay in IOC (>10% late)
Negligible	minimal to small reduction in technical performance, at detail level	irritating and awkward maintenance	consumption of some budget cushion	consumption of some slack—not on critical path

Session Analysis



Description Team scoring and classification are two activities which begin before the end of the group session and may continue as needed during the hour after it.

These activities are described in the following two sections.

Team Scoring

Objectives

to begin the team's evaluation of the individual risk statements by assigning probability and significance attributes

Who's in the Room?

SRE team

Duration

during the participants' scoring portion of the group session and for no more than 10 minutes after the participants have left the room

Preparation

The following must be completed prior to doing any scoring:

- An interview section of the group session is completed.
- Risk statements and context are captured.
- SRE team members revisit the project-specific definitions of impact and probability. (This is done during the participant's scoring at the end of the group session.)

Approach

Team scoring is nearly identical to the process used for participants scoring, except that the team members ***do not*** select their top five risk statements. Team scoring is led by the team leader.

Process

- Distribute scoring (evaluation) sheets for the group session.
- Review the project-specific definitions of the four levels of risk impact (negligible, marginal, critical, and catastrophic) that were determined with the project manager's help during contracting.
- Review the definitions of the three levels of risk probability: improbable, probable, and very likely.
- Each team member fills out the evaluation form for the session, ignoring the column for the top five risks.
- After team members (including the team leader) have written their assessment of risk impact and probability for each risk statement, collect the scoring sheets for the team's data compiler (typically, the

person on the team most adept at building and manipulating spreadsheets). At a more convenient time, ***but by the end of the day***, the tool operator enters each team member's values into the team members' scoring summary. If this is postponed, the amount of data will prohibit catching up later. This can best be done by two people: one to read the values and one to enter them.

The data compiler converts the scores that team members assign to each risk statement into risk exposure levels (from 1 to 6) using the risk scoring matrix agreed upon by the project manager during contracting. Within the spreadsheet, these risk exposures are evaluated across the team for mean (X-bar) and standard deviation of the sample (S), and the risk statements are then arranged in descending order by S.

Results

The final output of team scoring is a completed team members' scoring summary worksheet.

Key Considerations

Scoring should be done with the idea that the values assigned will change. As each group session is completed, the team learns more about the risks facing the program. Some risks that seemed very important in the early sessions will shrink in significance. Others will become more pertinent as time progresses. Remember that you will revisit these scores and that almost certainly, they will change when more data becomes available.

Tools

- electronic spreadsheet application
- notebook computer (full-size keyboard and mouse recommended)

Forms to Be Used

risk evaluation forms

Classification

Objectives to assign risk statements to elements of the SEI taxonomy

Who's in the Room? SRE team

Duration 30-40 minutes following the team scoring activity for each group session

Preparation Classification may be performed by using either the risk evaluation forms (around the conference table) or a wall chart and moving risk statement slips around. Such slips can usually be printed out in a suitable font size by the data compiler directly from the spreadsheet application being used to capture the risk statements

"Useful" Proximate Source The condition of a risk statement has many sources. In principle, there is only one *most proximate* source. All other nearby sources are simply "proximate sources."

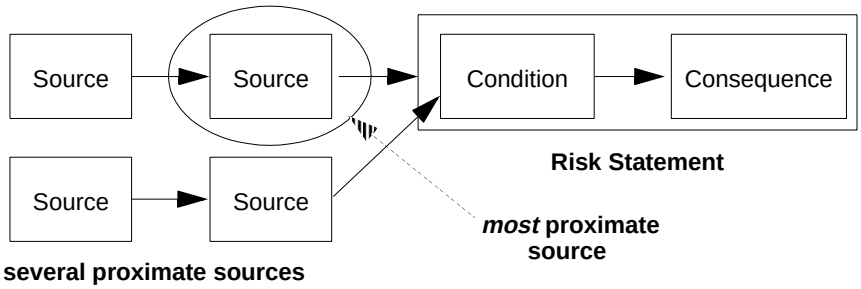


Figure 1: The Most Proximate Source

The *most* proximate source that the person being interviewed perceives may not be useful for classification purposes.

A "useful" proximate source is one that

- remains close enough to the original risk statement condition to be reasonably sure that if it had not happened, the condition would not exist

- is at a high enough level to suggest links to other risk statements
- can be acted on by the project manager

Project Manager's Control

Often, the useful proximate source of a condition in a risk statement is not in the project manager's control. This determination—whether the source ***is*** or ***is not*** within the project manager's sphere of control—starts the process of locating the risk statement in the taxonomy.

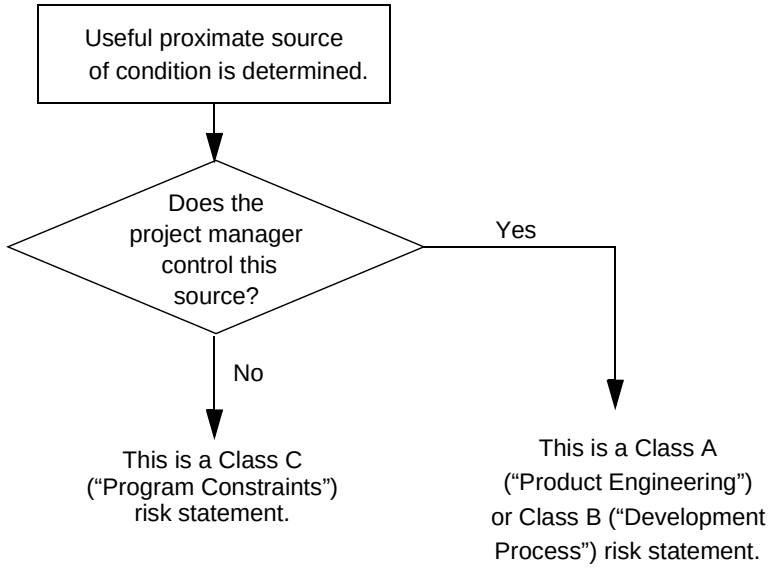


Figure 2: Locating the Risk Statement in the Taxonomy

Once it has been determined that the source of the condition is in the project manager's control, we must determine whether the source arises from one of the following:

- the nature of the product itself (Class A)
- the way the project is going about its development (Class B)

Approach

Classification is led by the team leader. The process is as follows:

1. Prepare slips with each risk statement from the group session (an unscored risk evaluation form works fine). Each slip should contain

the risk statement and the risk ID (e.g., “G2.6” is the sixth statement captured in the second group session).

2. Put up the taxonomy element wall chart and review “judgement call” criteria:
   - “useful” proximate sources
   - project manager’s control
3. Divide up risk statements among the team and have them place the statements under the elements on the wall chart that seem most appropriate, given the proximate source of the risk condition.
4. Ask the team to discuss the resulting classification and to then move risk statements around as seems appropriate (including risk statements from earlier sessions).
5. When all movement is completed, mark on each slip the letter/number of the taxonomy element that it ended up under (e.g., “A.5”).

At a more convenient time, **but by the end of the day**, the tool operator enters the taxonomic classification of each risk statement into the team members’ scoring summary. If this is postponed, the amount of data will prohibit catching up later. This can best be done by two people: one to read the values and one to enter them.

Results

The final output of classification is an agreed upon set of taxonomically classified risks statements for those risks captured in a group session.

Key Considerations

- Classification should be done with the idea that the values assigned may change. Remember that you will revisit these classifications and that they may change when more data becomes available.
- It may be convenient to hang the taxonomy wall chart or slips of paper with the class and element labels up on a wall in the meeting room. However, that chart should be ***covered*** while group session participants are in the room.
- It is very important that all risk statements generated during the day be classified and scored before the end of that same day. The extra effort to do so will pay dividends during the consolidation step.

Tools

- electronic spreadsheet application

- notebook computer (full-size keyboard and mouse recommended)
- slips of paper for each risk statement (cut up an unscored risk evaluation form from the session)

Forms to Be Used taxonomy element wall chart described on page 64

Taxonomic Group Definitions

This section provides the definitions of the taxonomic groups in the class, element, and attribute categories of the software development risk taxonomy. An overview of the taxonomy groups and their hierarchical organization is provided in Figure 1.

The taxonomy might be used to classify many different factors associated with the development of software-dependent systems such as development tasks, quality procedures, or sources or consequences of risk. However, the definitions as presented here are designed to facilitate the classification of the risks themselves, as associated with the development process.

NOTE: The material presented here is a reprint of Appendix B, Taxonomic Group Definitions, taken from the following technical report:

Carr, Marvin; Konda, Suresh; Monarch, Ira; Ulrich, Carol; & Walker, Clay. *Taxonomy Based Risk Identification* (CMU/SEI-93-TR-006, ADA266992). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

Taxonomy of Software Development Risks

A. Product Engineering

1. Requirement
a. Stability
b. Completeness
c. Clarity
d. Validity
e. Feasibility
f. Precedent
g. Scale

1. Design
a. Functionality
b. Difficulty
c. Interfaces
d. Performance
e. Testability
f. Hardware Constraints
g. Non-Developmental Software

1. Code and Unit Test
a. Feasibility
b. Testing
c. Coding/Implementation

1. Integration and Test
a. Environment
b. Product
c. System

1. Engineering Specialties
a. Maintainability
b. Reliability
c. Safety
d. Security
e. Human Factors
f. Specifications

B. Development Environment

1. Development Process
a. Formality
b. Suitability
c. Process Control
d. Familiarity
e. Product Control

1. Development System
a. Capacity
b. Suitability
c. Usability
d. Familiarity
e. Reliability
f. System Support
g. Deliverability

1. Management Process
a. Planning
b. Project Organization
c. Management Experience
d. Program Interfaces

1. Management Methods
a. Monitoring
b. Personnel Management
c. Quality Assurance
d. Configuration Management

1. Work Environment
e. Quality Attitude
f. Cooperation
g. Communication
h. Morale

C. Program Constraints

1. Resources
a. Schedule
b. Staff
c. Budget
d. Facilities

1. Contract
a. Type of Contract
b. Restrictions
c. Dependencies

1. Program Interfaces
a. Customer
b. Associate Contractors
c. Subcontractors
d. Prime Contractor
e. Corporate Management
f. Vendors
g. Politics

A Product Engineering

Product engineering refers to the system engineering and software engineering activities involved in creating a system that satisfies specified requirements and customer expectations. These activities include system and software requirements analysis and specification, software design and implementation, integration of hardware and software components, and software and system test.

The elements of this class cover traditional software engineering activities. They comprise those technical factors associated with the deliverable product itself, independent of the processes or tools used to produce it or the constraints imposed by finite resources or external factors beyond program control.

Product engineering risks generally result from requirements that are technically difficult or impossible to implement, often in combination with inability to negotiate relaxed requirements or revised budgets and schedules; from inadequate analysis of requirements or design specification; or from poor quality design or coding specifications.

A.1 Requirements

Attributes of the requirements element cover both the quality of the requirements specification and also the difficulty of implementing a system that satisfies the requirements.

The following attributes characterize the requirements element.

A.1-a. Stability

The stability attribute refers to the degree to which the requirements are changing and the possible effect changing requirements and external interfaces will have on the quality, functionality, schedule, design, integration, and testing of the product being built.

The attribute also includes issues that arise from the inability to control rapidly changing requirements. For example, impact analyses may be inaccurate because it is impossible to define the baseline against which the changes will be implemented.

A.1-b. Completeness

Missing or incompletely specified requirements may appear in many forms, such as a requirements document with many functions or parameters "to be defined"; requirements that are not specified adequately to develop acceptance criteria, or inadvertently omitted requirements. When missing information is not supplied in a timely manner, implementation may be based on contractor assumptions that differ from customer expectations.

When customer expectations are not documented in the specification, they are not budgeted into the cost and schedule.

A.1-c. Clarity

This attribute refers to ambiguously or imprecisely written individual requirements that are not resolved until late in the development phase. This lack of a mutual contractor and customer understanding may require re-work to meet the customer intent for a requirement.

A.1-d. Validity

This attribute refers to whether the aggregate requirements reflect customer intentions for the product. This may be affected by misunderstandings of the written requirements by the contractor or customer, unwritten customer expectations or requirements, or a specification in which the end user did not have inputs.

This attribute is affected by the completeness and clarity attributes of the requirements specifications, but refers to the larger question of the system as a whole meeting customer intent.

A.1-e. Feasibility

The feasibility attribute refers to the difficulty of implementing a single technical or operational requirement, or of simultaneously meeting conflicting requirements. Sometimes two requirements by themselves are feasible, but together are not; they cannot both exist in the same product at the same time.

Also included is the ability to determine an adequate qualification method for demonstration that the system satisfies the requirement.

A.1-f. Precedent

The precedent attribute concerns capabilities that have not been successfully implemented in any existing systems or are beyond the experience of program personnel or of the company. The degree of risk depends on allocation of additional schedule and budget to determine the feasibility of their implementation; contingency plans in case the requirements are not feasible as stated; and flexibility in the contract to allocate implementation budget and schedule based on the outcome of the feasibility study.

Even when unprecedented requirements are feasible, there may still be a risk of underestimating the difficulty of implementation and committing to an inadequate budget and schedule.

A.1-g. Scale

This attribute covers both technical and management challenges presented by large complex systems development.

Technical challenges include satisfaction of timing, scheduling and response requirements, communication among processors, complexity of system integration, analysis of inter-component dependencies, and impact due to changes in requirements.

Management of a large number of tasks and people introduces a complexity in such areas as project organization, delegation of responsibilities, communication among management and peers, and configuration management.

A.2 Design

The attributes of the design element cover the design and feasibility of algorithms, functions or performance requirements, and internal and external product interfaces. Difficulty in testing may begin here with failure to work to testable requirements or to include test features in the design.The following attributes characterize the design element.

A.2-a. Functionality

This attribute covers functional requirements that may not submit to a feasible design, or use of specified algorithms or designs without a high degree of certainty that they will satisfy their source requirements. Algorithm and design studies may not have used appropriate investigation techniques or may show marginal feasibility.

A.2-b. Difficulty

The difficulty attribute refers to functional or design requirements that may be extremely difficult to realize. Systems engineering may design a system architecture difficult to implement, or requirements analysis may have been based on optimistic design assumptions.

The difficulty attribute differs from design feasibility in that it does not proceed from pre-ordained algorithms or designs.

A.2-c. Interfaces

This attribute covers all hardware and software interfaces that are within the scope of the development program, including interfaces between configuration items, and the techniques for defining and managing the interfaces. Special note is taken of non-developmental software and developmental hardware interfaces.

A.2-d. Performance

The performance attribute refers to time-critical performance: user and real-time response requirements, throughput requirements, performance analyses, and performance modeling throughout the development cycle.

A.2-e. Testability

The testability attribute covers the amenability of the design to testing, design of features to facilitate testing, and the inclusion in the design process of people who will design and conduct product tests.

A.2-f. Hardware Constraints

This attribute covers target hardware with respect to system and processor architecture, and the dependence on hardware to meet system and software performance requirements. These constraints may include throughput or memory speeds, real-time response capability, database access or capacity limitations, insufficient reliability, unsuitability to system function, or insufficiency in the amount of specified hardware.

A.2-g. Non-Developmental Software

Since non-developmental software (NDS) is not designed to system requirements, but selected as a "best fit," it may not conform precisely to performance, operability, or supportability requirements.

The customer may not accept vendor or developer test and reliability data to demonstrate satisfaction of the requirements allocated to NDS. It may then be difficult to produce this data to satisfy acceptance criteria and within the estimated NDS test budget.

Requirements change may necessitate re-engineering or reliance on vendors for special purpose upgrades.

A.3 Code and Unit Test

Attributes of this element are associated with the quality and stability of software or interface specifications, and constraints that may present implementation or test difficulties.

A.3-a. Feasibility

The feasibility attribute of the code and unit test element addresses possible difficulties that may arise from poor design or design specification or from inherently difficult implementation needs.

For example, the design may not have quality attributes such as module cohesiveness or interface minimization; the size of the modules may contribute complexity; the design may not be specified in sufficient detail, requiring the programmer to make assumptions or design decisions during coding; or the design and interface specifications may be changing, perhaps without an approved detailed design baseline; and the use of developmental hardware may make an additional contribution to inadequate or unstable interface specification. Or, the nature of the system itself may aggravate the difficulty and complexity of the coding task.

A.3-b. Unit Test

Factors affecting unit test include planning and preparation and also the resources and time allocated for test.

Constituents of these factors are: entering unit test with quality code obtained from formal or informal code inspection or verification procedures; pre-planned test cases that have been verified to test unit requirements; a test bed consisting of the necessary hardware or emulators, and software or simulators; test data to satisfy the planned test; and sufficient schedule to plan and carry out the test plan.

A.3-c. Coding/Implementation

This attribute addresses the implications of implementation constraints. Some of these are: target hardware that is marginal or inadequate with regard to speed, architecture, memory size or external storage capacity; required implementation languages or methods; or differences between the development and target hardware.

A.4 Integration and Test

This element covers integration and test planning, execution, and facilities for both the contractual product and for the integration of the product into the system or site environment.

A.4-a. Environment

The integration and test environment includes the hardware and software support facilities and adequate test cases reflecting realistic operational scenarios and realistic test data and conditions.

This attribute addresses the adequacy of this environment to enable integration in a realistic environment or to fully test all functional and performance requirements.

A.4-b. Product

The product integration attribute refers to integration of the software components to each other and to the target hardware, and testing of the contractually deliverable product. Factors that may affect this are internal interface specifications for either hardware or software, testability of requirements, negotiation of customer agreement on test criteria, adequacy of test specifications, and sufficiency of time for integration and test.

A.4-c. System

The system integration attribute refers to integration of the contractual product to interfacing systems or sites. Factors associated with this attribute are external interface specifications, ability to faithfully produce system interface conditions prior to site or system integration, access to the system or site being interfaced to, adequacy of time for testing, and associate contractor relationships.

A.5 Engineering Specialities

The engineering specialty requirements are treated separately from the general requirements element primarily because they are often addressed

by specialists who may not be full time on the program. This taxonomic separation is a device to ensure that these specialists are called in to analyze the risks associated with their areas of expertise.

A.5-a. Maintainability

Maintainability may be impaired by poor software architecture, design, code, or documentation resulting from undefined or un-enforced standards, or from neglecting to analyze the system from a maintenance point of view.

A.5-b. Reliability

System reliability or availability requirements may be affected by hardware not meeting its reliability specifications or system complexity that aggravates difficulties in meeting recovery timelines. Reliability or availability requirements allocated to software may be stated in absolute terms, rather than as separable from hardware and independently testable.

A.5-c. Safety

This attribute addresses the difficulty of implementing allocated safety requirements and also the potential difficulty of demonstrating satisfaction of requirements by faithful simulation of the unsafe conditions and corrective actions. Full demonstration may not be possible until the system is installed and operational.

A.5-d. Security

This attribute addresses lack of experience in implementing the required level of system security that may result in underestimation of the effort required for rigorous verification methods, certification and accreditation, and secure or trusted development process logistics; developing to unprecedented requirements; and dependencies on delivery of certified hardware or software.

A.5-e. Human Factors

Meeting human factors requirements is dependent on understanding the operational environment of the installed system and agreement with vari-

ous customer and user factions on a mutual understanding of the expectations embodied in the human factors requirements. It is difficult to convey this understanding in a written specification. Mutual agreement on the human interface may require continuous prototyping and demonstration to various customer factions.

A.5-f. Specifications

This attribute addresses specifications for the system, hardware, software, interface, or test requirements or design at any level with respect to feasibility of implementation and the quality attributes of stability, completeness, clarity, and verifiability.

B. Development Environment

The development environment class addresses the project environment and the process used to engineer a software product. This environment includes the development process and system, management methods, and work environment. These environmental elements are characterized below by their component attributes.

B.1 Development Process

The development process element refers to the process by which the contractor proposes to satisfy the customer's requirements. The process is the sequence of steps—the inputs, outputs, actions, validation criteria, and monitoring activities—leading from the initial requirement specification to the final delivered product. The development process includes such phases as requirements analysis, product definition, product creation, testing, and delivery. It includes both general management processes such as costing, schedule tracking, and personnel assignment, and also project-specific processes such as feasibility studies, design reviews, and regression testing.

This element groups risks that result from a development process that is inadequately planned, defined and documented; that is not suited to the activities necessary to accomplish the project goals; and that is poorly communicated to the staff and lacks enforced usage.

B.1-a. Formality

Formality of the development process is a function of the degree to which a consistent process is defined, documented, and communicated for all aspects and phases of the development.

B.1-b. Suitability

Suitability refers to the adequacy with which the selected development model, process, methods, and tools support the scope and type of activities required for the specific program.

B.1-c. Process Control

Process control refers not only to ensuring usage of the defined process by program personnel, but also to the measurement and improvement of the process based on observation with respect to quality and productivity goals. Control may be complicated due to distributed development sites.

B.1-d. Familiarity

Familiarity with the development process covers knowledge of, experience in, and comfort with the prescribed process.

B.1-e. Product Control

Product control is dependent on traceability of requirements from the source specification through implementation such that the product test will demonstrate the source requirements. The change control process makes use of the traceability mechanism in impact analyses and reflects all resultant document modifications including interface and test documentation.

B.2 Development System

The development system element addresses the hardware and software tools and supporting equipment used in product development. This includes computer aided software engineering tools, simulators, compilers, test equipment, and host computer systems.

B.2-a. Capacity

Risks associated with the capacity of the development system may result from too few workstations, insufficient processing power or database storage, or other inadequacies in equipment to support parallel activities for development, test, and support activities.

B.2-b. Suitability

Suitability of the development system is associated with the degree to which it is supportive of the specific development models, processes,

methods, procedures, and activities required and selected for the program. This includes the development, management, documentation, and configuration management processes.

B.2-c. Usability

Usability refers to development system documentation, accessibility and workspace, as well as ease of use.

B.2-d. Familiarity

Development system familiarity depends on prior use of the system by the company and by project personnel as well as adequate training for new users.

B.2-e. Reliability

Development system reliability is a measure of whether the needed components of the development system are available and working properly whenever required by any program personnel.

B.2-f. System Support

Development system support involves training in use of the system, access to expert users or consultants, and repair or resolution of problems by vendors.

B.2-g. Deliverability

Some contracts require delivery of the development system. Risks may result from neglecting to bid and allocate resources to ensure that the development system meets all deliverable requirements.

B.3 Management Process

The management process element pertains to risks associated with planning, monitoring, and controlling budget and schedule; with controlling factors involved in defining, implementing, and testing the product; with

managing project personnel; and with handling external organizations including the customer, senior management, matrix management, and other contractors.

B.3-a. Planning

The planning attribute addresses risks associated with developing a well-defined plan that is responsive to contingencies as well as long-range goals and that was formulated with the input and acquiescence of those affected by it. Also addressed are managing according to the plan and formally modifying the plan when changes are necessary.

B.3-b. Project Organization

This attribute addresses the effectiveness of the program organization, the effective definition of roles and responsibilities, and the assurance that these roles and lines of authority are understood by program personnel.

B.3-c. Management Experience

This attribute refers to the experience of all levels of managers with respect to management, software development management, the application domain, the scale and complexity of the system and program, the selected development process, and hands-on development of software.

B.3-d. Program Interfaces

This attribute refers to the interactions of managers at all levels with program personnel at all levels, and with external personnel such as the customer, senior management, and peer managers.

B.4 Management Methods

This element refers to methods for managing both the development of the product and program personnel. These include quality assurance, configuration management, staff development with respect to program needs, and maintaining communication about program status and needs.

B.4-a. Monitoring

The monitoring includes the activities of obtaining and acting upon status reports, allocating status information to the appropriate program organizations, and maintaining and using progress metrics.

B.4-b. Personnel Management

Personnel management refers to selection and training of program members and ensuring that they: take part in planning and customer interaction for their areas of responsibility; work according to plan; and receive the help they need or ask for to carry out their responsibilities.

B.4-c. Quality Assurance

The quality assurance attribute refers to the procedures instituted for ensuring both that contractual processes and standards are implemented properly for all program activities, and that the quality assurance function is adequately staffed to perform its duties.

B.4-d. Configuration Management

The configuration management (CM) attribute addresses both staffing and tools for the CM function as well as the complexity of the required CM process with respect to such factors as multiple development and installation sites and product coordination with existing, possibly changing, systems.

B.5 Work Environment

The work environment element refers to subjective aspects of the environment such as the amount of care given to ensuring that people are kept informed of program goals and information, the way people work together, responsiveness to staff inputs, and the attitude and morale of the program personnel.

B.5-a. Quality Attitude

This attribute refers to the tendency of program personnel to do quality work in general and to conform to specific quality standards for the program and product.

B.5-b. Cooperation

The cooperation attribute addresses lack of team spirit among development staff both within and across work groups and the failure of all management levels to demonstrate that best efforts are being made to remove barriers to efficient accomplishment of work.

B.5-c. Communication

Risks that result from poor communication are due to lack of knowledge of the system mission, requirements, and design goals and methods, or to lack of information about the importance of program goals to the company or the project.

B.5-d. Morale

Risks that result from low morale range across low levels of enthusiasm and thus low performance, productivity or creativity; anger that may result in intentional damage to the project or the product; mass exodus of staff from the project; and a reputation within the company that makes it difficult to recruit.

C. Program Constraints

Program constraints refer to the “externals” of the project. These are factors that may be outside the control of the project but can still have major effects on its success or constitute sources of substantial risk.

C.1 Resources

This element addresses resources for which the program is dependent on factors outside program control to obtain and maintain. These include schedule, staff, budget, and facilities.

C.1-a. Schedule

This attribute refers to the stability of the schedule with respect to internal and external events or dependencies and the viability of estimates and planning for all phases and aspects of the program.

C.1-b. Staff

This attribute refers to the stability and adequacy of the staff in terms of numbers and skill levels, their experience and skills in the required technical areas and application domain, and their availability when needed.

C.1-c. Budget

This attribute refers to the stability of the budget with respect to internal and external events or dependencies and the viability of estimates and planning for all phases and aspects of the program.

C.1-d. Facilities

This attribute refers to the adequacy of the program facilities for development, integration, and testing of the product.

C.2 Contract

Risks associated with the program contract are classified according to contract type, restrictions, and dependencies.

C.2-a. Type of Contract

This attribute covers the payment terms (cost plus award fee, cost plus fixed fee, etc.) and the contractual requirements associated with such items as the Statement of Work, Contract Data Requirements List, and the amount and conditions of customer involvement.

C.2-b. Restrictions

Contract restrictions and restraints refer to contractual directives to, for example, use specific development methods or equipment and the resultant complications such as acquisition of data rights for use of non-developmental software.

C.2-c. Dependencies

This attribute refers to the possible contractual dependencies on outside contractors or vendors, customer-furnished equipment or software, or other outside products and services.

C.3 Program Interfaces

This element consists of the various interfaces with entities and organizations outside the development program itself.

C.3-a. Customer

The customer attribute refers to the customer's level of skill and experience in the technical or application domain of the program as well as difficult working relationships or poor mechanisms for attaining customer agreement and approvals, not having access to certain customer factions, or not being able to communicate with the customer in a forthright manner.

C.3-b. Associate Contractors

The presence of associate contractors may introduce risks due to conflicting political agendas, problems of interfaces to systems being developed by outside organizations, or lack of cooperation in coordinating schedules and configuration changes.

C.3-c. Subcontractors

The presence of subcontractors may introduce risks due to inadequate task definitions and subcontractor management mechanisms, or to not transferring subcontractor technology and knowledge to the program or corporation.

C.3-d. Prime Contractor

When the program is a subcontract, risks may arise from poorly defined task definitions, complex reporting arrangements, or dependencies on technical or programmatic information.

C.3-e. Corporate Management

Risks in the corporate management area include poor communication and direction from senior management as well as non-optimum levels of support.

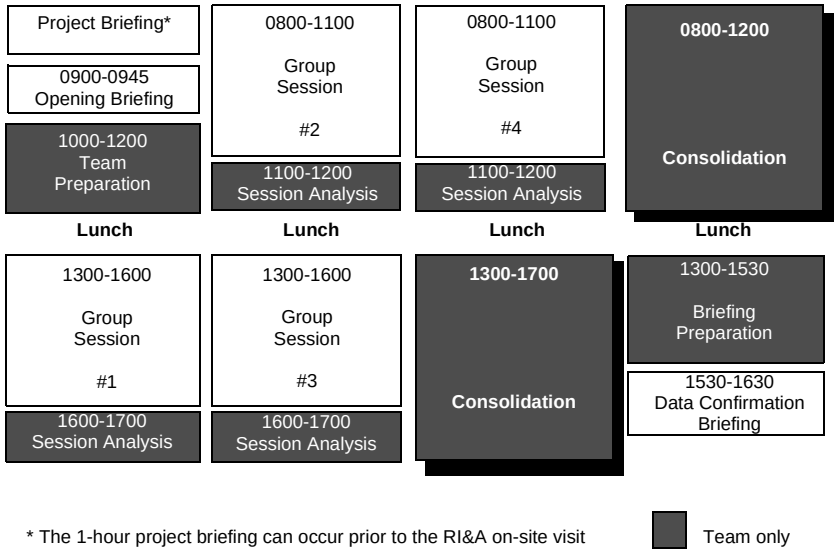
C.3-f. Vendors

Vendor risks may present themselves in the forms of dependencies on deliveries and support for critical system components.

C.3-g. Politics

Political risks may accrue from relationships with the company, customer, associate contractors or subcontractors, and may affect technical decisions.

Consolidation



Objectives

- to bring together and interpret the information generated during the group sessions and the team scoring and classification sessions
- to prepare the SRE team to produce the data confirmation briefing slides

Who's in the Room?

The entire SRE team is involved in consolidation. Some tasks may be assigned to subgroups within the team; every SRE team member does not need to be part of every step.

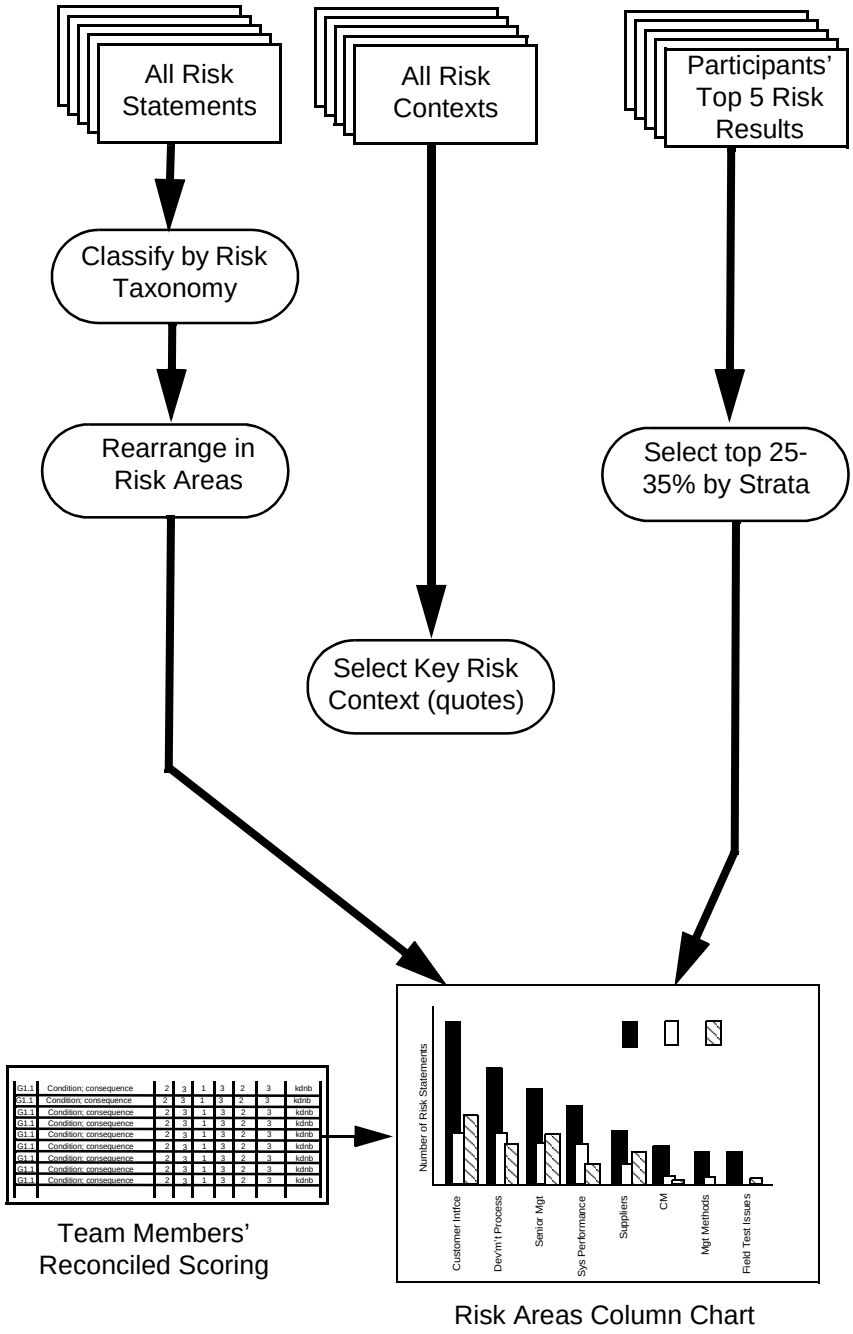
Tasks During Consolidation

The diagram on the next page shows the tasks to be completed during consolidation.These tasks include:

- reconcile scoring
- rearrange risk statements into risk areas
- determine participants' top risks
- select key risk context
- aggregate data

Each task is described in the sections that follow.

The Overall Consolidation Process



Reconcile Scoring

Objective

- to generate the team's consensus on the most important risks to the project
- to create a complete ranking of all risks

Who's in the Room?

The entire SRE team must be involved in the reconciliation of scores.

Duration

one hour following the completion of the last group session and team scoring and classification steps

Preparation

The following must be completed prior to doing reconciliation:

- All risks have been scored by team members.
- Within the spreadsheet, these risk exposures have been evaluated across the team for **mean** (X-bar) and **standard** deviation of the sample (S), and the risk statements have been arranged in descending order by S. This produces the team members' scoring summary form.
- Context notes from each of the group sessions have been photocopied (or printed), distributed to each team member, and read.

Approach

Scoring reconciliation is conducted by the team leader using the following process:

1. The data compiler prints and distributes the team members' scoring summary to all team members.
2. Beginning from the top of the list—with the risk statement for which the risk exposure values given by team members were in the greatest disagreement—count down the list and draw a line which demarcates the top 25-35% of the risks. This will be the goal end point for the process.

3. Begin the discussion with the risk statement at the top of the list. Have the person giving the highest risk exposure value and the one giving the lowest value explain their rationales to the others.
4. Allow the discussion to proceed as other team members become involved. When the discussion appears to be approaching either consensus or intractable differences, end the discussion and poll each member to either provide a revised risk exposure value or state that they are "standing pat." Note: these new risk exposure values can be determined directly, without revising the original impact and probability values.
5. Allow no more than one hour for this process, terminating when either the time period has expired or the 25-35% line has been reached.

The data compiler can use the process below to determine the final scores and the list of the team's "most important risks" without input from the rest of the SRE team.

1. After entering all the revised risk exposure values, the data compiler re-sorts the list of risk statements in descending order by the mean of the team's risk exposure values.
2. The data compiler scans the list again from the top to find a point in the range of the top 25-35% risk statements at which a clear breaking point in the means occurs. The risk statements above this breaking point are declared the SRE team's most important risk statements.
3. Each score on the list is then rounded to the nearest whole number from 1 to 6. This is the final risk exposure value to be given to the project manager. Note: This is to preserve the sense that the risk exposure values are ordinal numbers, not points on a continuous, linear scale.

Results

The output of scoring analysis and reconciliation is the team's reconciled scoring - the ranked list of risks faced by the project.

Key Considerations

- This process must be done as quickly as possible; keep arguments concise and impersonal.
- Maintain focus on the risk statements.

- Use context to stimulate discussions.
- Keep in mind the project-specific definitions for risk impact and the definitions of probability.

Tools laptop computer with electronic spreadsheet application

Forms to Be Used The team members' scoring worksheet and the team's reconciled scoring form are used. Samples of these forms are provided on the following pages.

Team Members' Scoring Worksheet

	Risk No	Risk Statement	sgb				gjp				wrw				rew				Risk Exp			
			1	P	RE	Rev RE	I	P	RE	Rev RE	1	P	RE	Rev RE	1	P	RE	Rev RE	Mx	Mn	S	Mean
G3.11	6	There are rumors that the telephone company is unhappy with the Screen Display design and see it as representative of S31 work. They may cancel the project.	1	1	**1**		1	1	**1**		4	2	**5**		4	1	**4**		**5**	**1**	**2.06**	**2.75**
G2.11	20	Concerned about configuration management between development and field test sites; lack of CM may cause version mismatches, lost time, and rework.	3	2	**4**		4	3	**6**		3	2	**4**		1	1	**1**		**6**	**1**	**2.06**	**3.75**
G3.06	18	VP introducing new system requirements without budget or schedule relief; this is muddying the protect's lines of authority.	2	1	**2**		4	3	**6**		1	3	**3**		1	2	**2**		**6**	**2**	**1.89**	**3.25**
G4.11	46	Toivolia accounting department wanted to do this job, and they are still trying to prove they could do it better; delay in approval cycles, have to constantly prove S31's solution is "best."	1	1	**1**		4	2	**5**		2	3	**4**		2	3	**4**		**5**	**1**	**1.73**	**3.5**
G3.17	16	There is a perception that upper management arbitrarily revised the project cost estimate downward to win the contract; people may give up trying to meet deadlines and performance bogeys.	1	1	**1**		3	2	**4**		3	1	**3**		3	3	**5**		**5**	**1**	**1.71**	**3.25**
G3.04	64	There are no procedures or processes in place to enforce CM; delays, time spent testing the wrong system.	4	2	**5**		2	2	**3**		3	3	**5**		2	1	**2**		**5**	**2**	**1.5**	**3.75**
G1.16	58	the past history of this company is that code and design are poorly documented; there may be difficulty in maintaining what is supposed to be a "flagship" product.	2	1	**2**		2	3	**4**		3	3	**5**		1	2	**2**		**5**	**2**	**1.5**	**3.25**
G3.14	36	The three-letter algorithm may result in so many pages of possibles (e.g., for "SMI") that operators may get frustrated and refuse to use the system.	2	1	**2**		1	2	**2**		3	2	**4**		3	3	**5**		**5**	**2**	**1.5**	**3.25**
G1.08	41	Acceptance configuration of the system does not replicate the actual operational system configuration; unpredictable consequences and rework in the field.	3	3	5		2	2	3		4	3	6		4	3	6		6	3	1.41	5
G3.05	14	The VP is undercutting the project manager and introducing new requirements; these may remain hidden, and no test cases will be developed for them.	4	2	5		4	3	6		2	3	4		2	2	3		6	3	1.29	4.5
G1.01	57	Requirements are changing because of outside influences (vice president); this will affect quality of the code, integration, morale, and schedule.	3	3	5		4	3	6		2	2	3		2	3	4		6	3	1.29	4.5
G4.09	52	The C++ compiler may not perform adequately; might have to be replaced, for which there is no budget, and schedule impact due to new learning curve.	3	2	4		2	1	2		3	3	5		3	1	3		5	2	1.29	3.5
G2.10	62	Conditions during field startup (testing at night) may mean that our best integrators & testers will not be willing to go; troubleshooting may require excessive time.	3	2	4		3	3	5		2	2	3		2	1	2		5	2	1.29	3.5
G3.10	13	There are rumors that low performers in the project may get fired to serve as a lesson to the rest, so many people are job hunting; we may not have everyone we need to meet our deadlines.	3	2	4		1	1	1		2	2	3		2	1	2		4	1	1.29	2.5
G4.04	40	Upper management has not approved C++ training for project staff—the needed training may have to come from project budget; profit will be in jeopardy.	1	2	2		3	3	5		2	3	4		1	3	3		5	2	1.29	3.5

Team's Reconciled Scoring

	Risk No	Risk Statement	sgb				gjp				wrw				rew				Revised Risk Exp.					
			1	P	RE	Rev RE	I	P	RE	Rev RE	1	P	RE	Rev RE	1	P	RE	Rev RE	Mx	Mn	S	Mean	Final RE	Team Top Risk?
G1.08	41	Acceptance configuration of the system does not replicate the actual operational system configuration; unpredictable consequences and rework in the field.	3	3	**5**	**5**	2	2	**3**	**5**	4	3	**6**	**6**	4	3	**6**	6	**6**	**5**	**0.577**	**5.5**	6	Yes
G2.09	45	The C++ compiler has bugs; added time to develop workarounds, aggravates lack of C++ experience of developers, may have to replace compiler, for which there is no budget.	4	3	**6**	**6**	3	3	**5**	**5**	4	2	**5**	**6**	3	3	**5**	5	**6**	**5**	**0.5**	**5.25**	5	Yes
G1.09	33	We've never tried to make 10 computers work together like this; we don't know what we don't know; could delay final system acceptance.	4	3	**6**	**6**	4	2	**5**	**5**	3	2	**4**	**4**	4	3	**6**	6	**6**	**4**	**0.957**	**5.25**	5	Yes
G1.06	43	Have to support 50 terminals on each computer with 3-second response time, but have only tested with 25; might have to buy more computers, network overhead, electronic switch might be affected.	2	2	**3**	**5**	3	3	**5**	**5**	3	3	**5**	**5**	4	3	**6**	6	**6**	**5**	**0.5**	**5.25**	5	Yes
G1.13	44	No performance analysis has been done for the system; we don't know what we don't know.	4	2	**5**	**5**	4	3	**6**	**6**	4	2	**5**	**5**	3	3	**5**	5	**6**	**5**	**0.5**	**5.25**	5	Yes
G1.07	27	Our programmers are FORTRAN programmers; it's going to be a tough learning curve to move to C++, may cause delays, rework, hard-to-find bugs.	3	3	**5**	**5**	3	3	**5**	**5**	3	3	**5**	**5**	3	3	**5**	**5**	**5**	**5**	**0**	**5**	5	Yes
G3.02	39	Developers are working from their own interpretation of requirements documents, not using the developed test scenarios; the system may not be properly tested and may fail final acceptance—alternatively, lots of rework.	3	3	**5**	**5**	3	3	**5**	**5**	4	2	**5**	**5**	3	2	**4**	4	**5**	**4**	**0.5**	**4.75**	5	Yes
G3.05	14	The VP is undercutting the project manager and introducing new requirements; these may remain hidden, and no test cases will be developed for them.	4	2	**5**	**5**	4	3	**6**	**6**	2	3	**4**	**4**	2	2	**3**	4	**6**	**4**	**0.957**	**4.75**	5	Yes
G1.01	56	Requirements are changing because of outside influences (vice president); this will affect quality of the code, integration, morale, and schedule	3	3	**5**	**5**	4	3	**6**	**6**	2	2	**3**	**4**	2	3	4	4	6	4	0.957	4.75	5	Yes
G2.13	19	Conflicts with the customer are not being resolved in a timely manner; a lot of unplanned time spent educating the customer, drag on the schedule.	3	2	**4**	**4**	4	3	**6**	**6**	2	3	**4**	**4**	3	2	4	4	6	4	1	4.5	5	Yes
G1.03	28	No impact analysis of changed requirements is being done; may wind up with conflicting features, goals, and requirements.	2	3	**4**	**4**	4	3	**6**	**6**	3	2	**4**	**4**	2	3	4	4	6	4	1	4.5	5	Yes
G3.15	57	The effect of loading on the network was considered to be "negligible" — no tests were done. One computer may handle 50 operators OK, but 10 computers may not be able to handle 500 operators.	3	2	**4**	**4**	3	2	**4**	**4**	4	2	**5**	**5**	3	3	5	5	5	4	0.577	4.5	5	Yes
G2.06	50	There is no formal change control process that coordinates all affected groups; test plans are not keeping up with changes.	2	3	**4**	**4**	3	2	**4**	**4**	3	3	**5**	**5**	2	3	4	4	5	4	0.5	4.25	4	Yes
G2.05	42	Requirements seem to be changing; can't be sure that the test cases cover all requirements.	2	2	**3**	**3**	3	3	**5**	**5**	2	3	**4**	**4**	3	3	5	5	5	3	0.957	4.25	4	Yes

Rearrange Risk Statements into Risk Areas

Objectives to arrange the risk statements into risk areas - groups of risks that can be mitigated together

Who's in the Room? SRE team

Duration one hour following the completion of the classification step

Preparation The classification step must be completed prior to rearranging risk statements.

Approach The team leader leads the following process.

1. Beginning with the risk statement slips in their taxonomic arrangement from the previous step, remove the labels and consider the risk statements for groups that could be mitigated with the same general approach.
2. Ask each team member to begin silently moving risk statements around into new clusters.
3. When movement has slowed, ask team members to provide labels of one to three words for each cluster and then discuss each label. The labels should be written on slips of paper and placed above the clusters.
4. Continue moving, clustering, and labeling until the clusters have been reduced to a reasonable number (seven to eleven). These are the risk areas.
5. In the rare instance where a risk statement falls under two different risk areas and the group cannot decide where to put it, the risk statement can be duplicated and a copy placed under each label. ***Avoid this wherever possible***—having duplicate risk statements will complicate consolidation and the construction of the Data Confirmation Briefing. If more than two risk statements are duplicated, reassess the risk area structure to make the duplication unnecessary.

6. Gather the risk statements up in their clusters, with the labels on top of each cluster, and give them to the team's data compiler to record both the taxonomy element and the risk area name beside each risk statement.
7. The data compiler reports the final count of risk statements in each risk area and prepares to generate the risk areas column chart as shown on page 102.

Results

The outputs are risk areas (clusters of risk statements that can be mitigated as a group).

Key Considerations

- Keep asking yourself, "What makes all of these risk statements that can be mitigated together?"
- It's okay to have a risk area with only one statement in it, but be prepared with solid logic about why this was necessary.
- Creating more than eleven risk areas should be avoided above all else, and having 5 to 9 risk areas is highly desirable. (This is because larger numbers of risk areas become difficult for anyone to comprehend—to remember, focus on, or prioritize. Also, the number of relationship analyses that need to be made between risk areas during the Interrelationship Digraph process increases dramatically as you go up in risk areas (e.g., 36 analyses for 9 risk areas, 45 analyses for 10 risk areas, 55 analyses for 11 risk areas 66 analyses for 12 risk areas), so more risk areas create more unnecessary work for the team.)

Logistics

- You'll need a large work space that the whole team can see at the same time. A big, open wall or whiteboard that the risk statements can be taped to has worked best in the past.
- Printing the risk statements in a large font will help everyone to read them at a distance.

Forms to Be Used

None.

Determine Participants' Top Risks

Objectives

to generate the participants' view of the most important risks to the project

Who's in the Room?

SRE Team. This step may be done by a subgroup of the team—it is a purely mechanical process that does not require decision making or consensus.

Duration

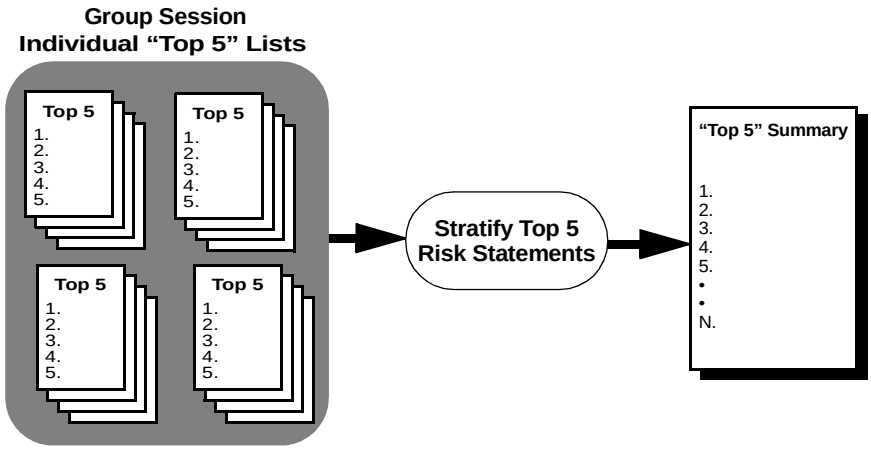
30 minutes after the rearrangement of risk statements into risk areas

Preparation

No preparation is required for this step; it may be done any time after the group sessions are completed.

Approach

This process is shown graphically below:



1. Determine how many risk statements constitute 25% of the total number of risk statements identified.

2. Select the risk statements from the individual "Top Five" lists in rounds, beginning with each participant's top risk, and proceeding to each participant's second risk, and so on, ***in full rounds***, until either the 25% figure has been passed, or all the participants' selections have be exhausted

   If an individual's choice is already on the list (selected earlier by another), nothing changes. Move on to the next individual.

   Note: The absolute number of risks selected using this method cannot be determined in advance. It depends on the number of participants in the interviews and the extent to which they agree with one another as to which risk statements represent the "most important risks to the program."

3. Give the results to the team's data compiler, who then maps the risk statements into the risk areas and summarizes how many are in each area.

Results

The output of this activity is a list of the most important risks to the project as viewed by the participants.

Points to Remember

- This is a mechanical process and can be done at any time prior to consolidation.
- It only provides insight into the risk statements the interviewees perceived as "most important" within the group session. It includes no perspective on risk statements from other group sessions.

Logistics

There are no special requirements. The process can be done on a laptop, flipchart, or a piece of paper.

Forms to Be Used

The completed group session evaluation forms are used as input; no special form is used for output.

Select Key Risk Context

Objectives

to preserve the immediacy and personal perspective of the interview when reporting on risk issues

Who's in the Room?

SRE team

Duration

30 minutes following the rearrangement of risk statements into risk areas.

Preparation

The following must be completed prior to selecting risk context:

- classification
- rearrangement of risk statements into risk areas.

Approach

For each identified risk area, one or two team members review the risk statements and their associated context for particularly vibrant metaphors and descriptions of the concern or issue said by participants during the interview. Examples include

- "project death spiral"
- "We're playing liar's poker here."
- "I'm afraid we may break through the ice out at Toivolia in the middle of acceptance testing."
- "The computer's thrashing itself to death."
- "They keep talking as if the system should work like Lotus 1-2-3, or like a video game."

When the key pieces of context that support a risk area have been highlighted, they are given to the team for use during the preparation of slides for the data confirmation briefing.

Caution: Avoid expressions that seem to be unique to an individual (to avoid implicit attribution). Look for phrases heard often during the interviews, or particularly picturesque language that is widely used in the industry.

Results The outputs are key context phrases that can be used in the preparation of the data confirmation briefing slides.

Key Considerations Be careful to preserve confidentiality. Make sure that the colorful context you pick is not a “stock phrase” already well associated with that individual.

Logistics This process only requires a private work area, a table to work around, and copies of the session records from all group sessions.

Forms to Be Used No special forms are required.

Aggregate Data

Objectives to complete the final consolidation activity - aggregate the relevant data in the form of a column chart

Who's in the Room? SRE team

Duration 30 minutes following the rearrangement of risk statements into risk areas.

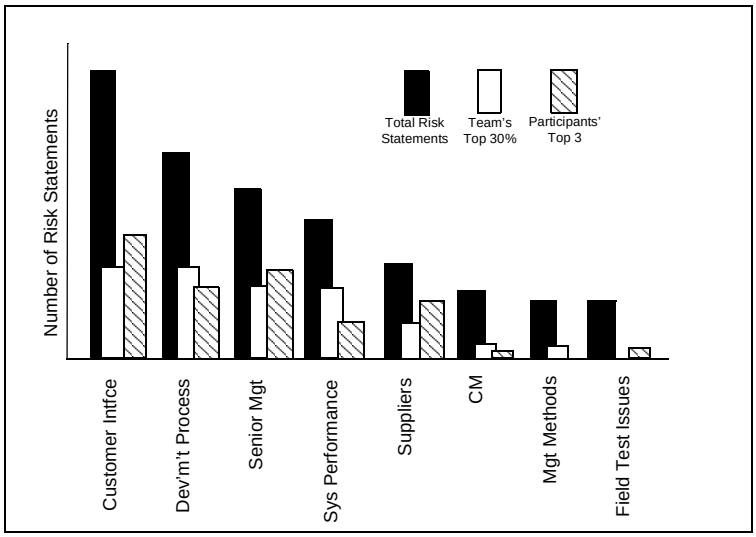
Preparation All prior consolidation steps must be completed prior to doing any aggregation of data.

Approach A straight-forward column chart is constructed to compare the total number of risk statements in each risk area with the following:

- the number of those statements judged by the team to be among the top 25-35% in terms of risk exposure
- the number of those risks viewed by the participants themselves as among the most important risks to the program

An example of such a chart is shown below:

Risk Areas Column Chart



Results

The output is a column chart that can be used in the preparation of the data confirmation briefing slides.

Key Considerations

- This column chart has long been the centerpiece of the SRE Data Confirmation Briefing. Make sure that each SRE team member understands what the chart says and what it *does not* say.
- The most important message of the chart is the number of risk areas that the SRE team found.
- The second most important message is how many risk statements were grouped into each risk area.
- The third most important message is how many of those risk statements in each risk area were considered to represent potentially serious risks to the project. For this there are two perspectives: the SRE team's and the interview participants'.
- Although this graphic can be set up in an electronic spreadsheet template in advance, it is good risk management to have a team member on hand who knows how to create such a graphic in an electronic spreadsheet program from scratch.

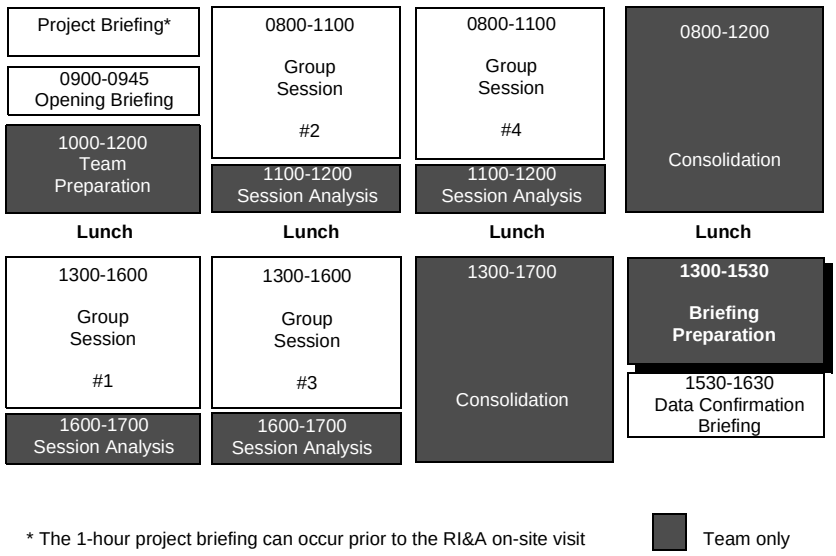
Logistics

This task can be done by one person using an electronic spreadsheet with graphical output. It requires only simple data manipulation. The most difficult task may be the mechanics of importing the graphic from the spreadsheet into the presentation slide.

Forms to Be Used

No special form is required. An electronic spreadsheet template for this graphic can easily be created in advance, using dummy information.

Data Confirmation Briefing Preparation



Objectives to create the data confirmation briefing presentation materials

Who's in the Room? SRE team

Duration 2 hours

Preparation The following must be completed prior to creating the data confirmation briefing:

- all group sessions
- all context review
- all team analysis, scoring, and reconciliation
- classification of risks
- consolidation

Approach

Presenting the data confirmation briefing should take a half-hour or less.

At its simplest level, this briefing is a presentation of the data that was collected in an organized fashion. The organizing structure is the risk area listing. The team leader leads development of the results briefing presentation which should include the following:

- "boilerplate" cover page
- review of the SRE process
- list of risks and their attributes
- risk classification results
- "top n" list of risks
- description of "next steps" the organization should take
- placeholder for project manager's closing comments

After the presentation has been created do the following:

- Make transparencies of the slides.
- Make a hard copy of the slides for the project manager.
- Make a dry run of the presentation.

Sample Data Confirmation Briefing Outline

The following outline presents an example of the data confirmation briefing.

Item	Description
Boilerplate cover page	• sets the stage • a place for program manager to introduce the team leader • time for team leader's introductory comments
SRE objectives	overall objectives of an SRE
SRE process overview	shows the larger context into which this RI&A effort fits
RI&A process	• schedule of work sessions for the participants and team members ("where we've been") • RI&A process flowchart ("what we've been through")

Item	Description
summary of activities	numbers: how many sessions, how many participants, how many risk statements, and so on
summary of findings	• risk area names • risk statements by risk area (risk areas column chart) • summary analysis of team and participant scores
findings by risk area	• observations for each area • direct quotes and risk statements, as appropriate
next steps	• interim report: why and when • MSP: when and how

Results

The results are the following data confirmation briefing presentation materials:

- transparencies of slides
- hard copy of slides for the project manager

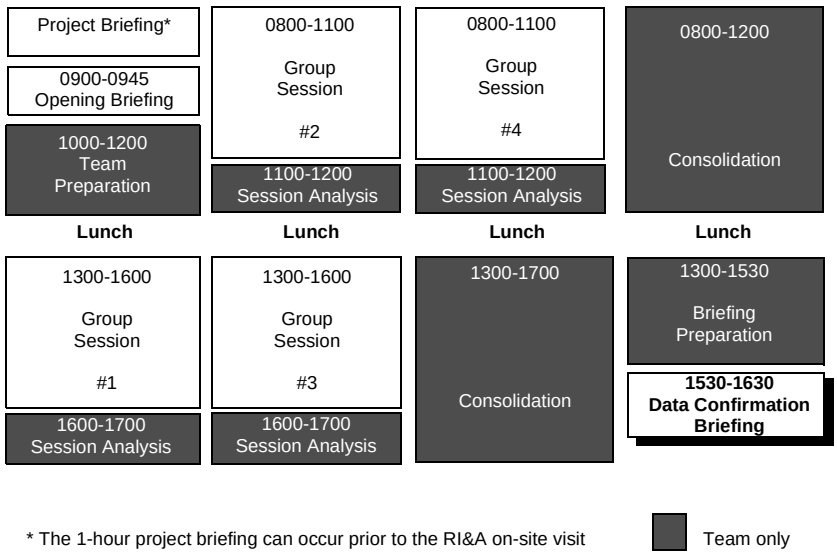
Key Considerations

Remind participants of the non-attribution and confidentiality principles.

Logistics

It is best to have a direct display device to make this presentation directly from the slide presentation software. If this is not possible, quick access to a photocopier for creating transparencies and making a hard copy for the project manager becomes essential.

Data Confirmation Briefing



Objectives

to present the project with the results of the Risk Identification and Analysis (RI&A) phase of the SRE

Who's in the Room?

- project manager
- all participants
- any other project members the project manager chooses to invite
- SRE team

Duration

30 minutes

Preparation

Prior to giving the data confirmation briefing, the following must be accomplished:

- Presentation transparencies and a hard copy for the project manager have been prepared.

- the project manager and all participants are in attendance.

Approach

The presentation is a formal briefing during which the following occurs:

- The project manager introduces the team leader.
- The team leader presents the data confirmation briefing.
- After the presentation, the team leader invites the project manager to comment.
- The project manager shares comments with the audience.
- The team leader gives a hard copy of the presentation to the project manager.

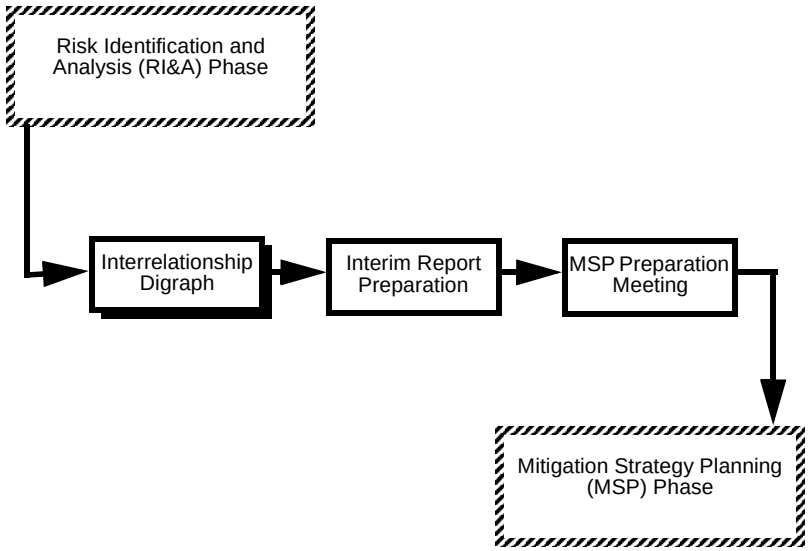
Results

The result is the official ending to the RI&A phase of the SRE.

Points to Remember

Participants need to see their manager introduce the team leader before the briefing, and summarize the importance of risk management to the project at the end of the briefing.

Interrelationship Digraph



Objectives

- to explore the relationships among risk areas identified during the Risk Identification and Analysis phase of the SRE
- to identify risk areas that include conditions which are creating similar conditions in other risk areas—irrespective of the rated "importance" of those risks—so that the SRE team and project manager can consider whether those "root-cause" risk areas should be mitigated first during the Mitigation Strategy Planning (MSP) phase

Who's in the Room?

the SRE team or a subteam taken from it

Duration

The digraph should take one hour (but only after being away from the data for a day or two)

Documentation of the results and an interpretive analysis of them may take several hours to days, depending on the complexity and sensitivity of the conclusions reached. For example, if the results point to a lack of commitment from management above the project having the SRE (not an uncommon occurrence), it is advisable to re-examine the interrelation-

ships and carefully explain (in writing) why the data support this conclusion. This should also be reviewed with as many other members of the SRE team as is practicable, to make sure that it is a conclusion that everyone can support. Do not shortcut this part of the process.

Source of Process

This description of an interrelationship digraph building process is taken from the *Continuous Risk Management Guidebook*, Chapter A-14 (beginning at page 345), and that chapter should be used as the reference for the general process. The following description is consistent with the reference, though not as detailed, and has been modified slightly to address the particular issues of the SRE team at this point in the SRE process.

Preparation

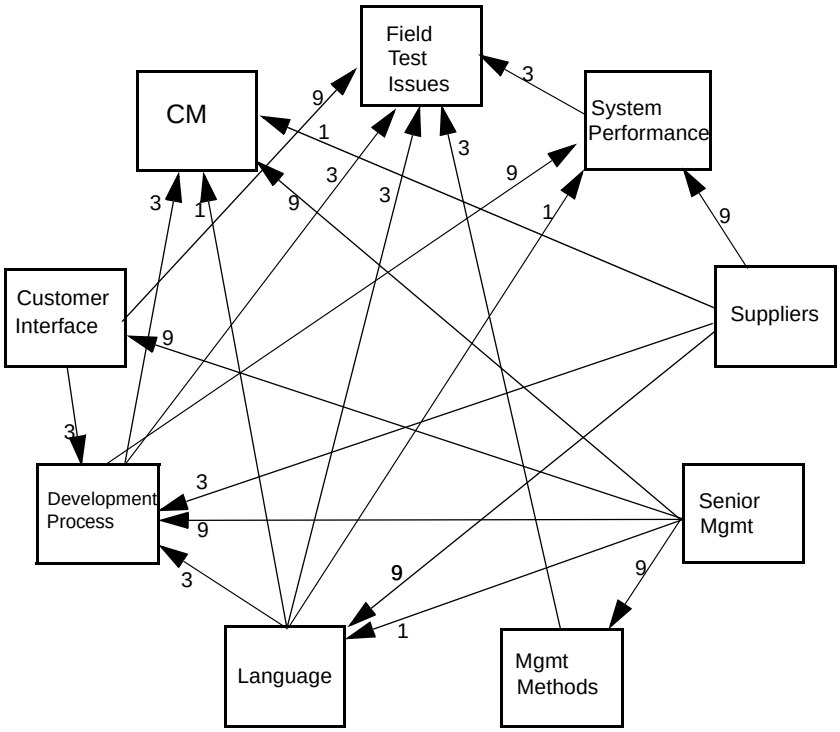
Follow the steps below to prepare for the construction of the interrelationship digraph:

1. Arrange all the risk statements in their risk areas, ideally using a single sheet of paper for each risk area, with the risk area label in large letters at the top of the page. Include with each risk statement the final risk exposure values determined by the team (see Reconcile Scoring on page 91) and identify the participants' top risks. Make a copy of these risk area sheets for each team member who will be participating in the interrelationship digraph building session.
2. Reproduce the session recorder notes (context) from each of the sessions and have a full copy of this context available for each of those participating in the interrelationship digraph building session.
3. Make reduced-size copies of the risk area sheets (complete with all the risk statements belonging to that area) that are small enough to be taped on a large whiteboard in a roughly circular layout, using all the whiteboard space that is available to do so.

Approach The following table describes how to construct an interrelationship digraph for an SRE after having taped the listings of risk statements by risk area on a whiteboard:

Step	Action
1	**Draw a relationship arrow between any two risk areas.** Look at one pair of risk areas and determine, by consensus, if there is an interrelationship between the risk statements. Do the conditions embodied in the statements of risk area X cause or influence conditions embodied in the statements of risk area Y? If yes, draw an arrow from risk area X to risk area Y. *Be very careful to make the evaluation on the basis of the relationship between the risk statements in the two risk areas, not on the basis of the name of the risk areas.* It is a common temptation to load more meaning into the name of the risk area (e.g., "Requirements" or "Senior Management") than can be supported by the risk statements that were captured in the interviews and collected under that label.
2	**Apply a weighting factor to the arrow.** Determine whether the relationship is "significant" (weighting factor of 9), "medium" (3), or "weak" (1).
3	**Repeat steps 1 and 2 for every pair of risk areas.** Proceeding around the circle of risk areas systematically, be sure that every pair of areas has been evaluated for an interrelationship, and that all interrelationships have been assigned a weighting factor of 1, 3, or 9.
6	**Review and revise, as necessary.** After comparing every pair of risk areas, review the relationships and make any necessary changes.
7	**Tally arrow information.** Count and record the number of incoming and outgoing arrows for each risk area. Calculate and record the total weight for each risk area (the sum of weights of all the arrows going into or out of the area).
8	**Select key items.** Use the tallied arrow information, experience, and judgement to reach consensus on the key risk areas to be worked on. Generally these should be the areas with the largest number of outgoing arrows (risk areas that predominantly include "Cause/Driver" risk statements) and the highest total weight.

Results A typical interrelationship digraph for an SRE and its results matrix are shown below.

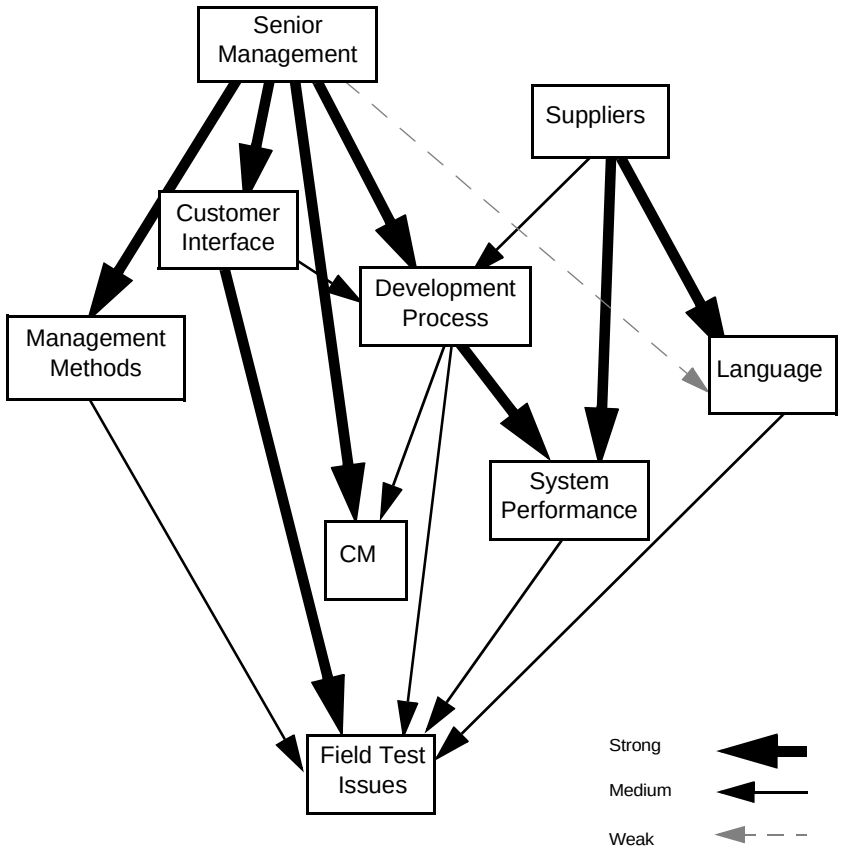


	Field Test Issues	System Performance	Suppliers	Senior Management	Management Methods	Language	Development Process	Customer Interface	CM	Cause/ Driver ⇑	Result/ Rider ⇐	Total Weight
Field Test Issues	•	3⇐			3⇐	3⇐	3⇐	9⇐		0	5	21
System Performance	3⇑	•	9⇐			1⇐	9⇐			1	3	22
Suppliers		9⇑	•			9⇑	3⇑		1⇑	4	0	22
Sr. Mgmt				•	9⇑	1⇑	9⇑	9⇑	9⇑	5	0	37
Mgt Methods	3⇑			9⇐	•					1	1	12
Language	3⇑	1⇑	9⇐	1⇐		•	3⇑		1⇑	4	2	18
Development Process	3⇑	9⇑	3⇐	9⇐		3⇐	•	3⇐	3⇑	3	4	33
Customer Interface	9⇑			9⇐			3⇑	•		2	1	21
CM			1⇐	9⇐		1⇐	3⇐		•	0	4	14

Interpreting the Results

The interrelationship digraph results can be redrawn in a way that more clearly identifies the important interrelationships and the risk areas that deserve first consideration as candidates for mitigation strategy planning. This is called an “interrelationship hierarchy” because the risk areas higher on the chart have risk statement conditions which are closer to

"root causes" than those lower on the chart. The example shown below corresponds to the interrelationship digraph in the previous section..



Interrelationship Hierarchy

Notice that some of the "weak" interrelationships of the interrelationship digraph have been removed from the depiction above, particularly when the effect is covered by a two-step relationship. For example, if A strongly affects B, which has a medium effect on C, but A also weakly affects C directly, it is reasonable to eliminate the weak effect of A on C from the hierarchical depiction, since it probably adds no new insight.

The interrelationship hierarchy can be a powerful and easily-grasped tool for explaining why one risk area should be attacked before another. In the example digraph above, the team would argue that the risk areas "Senior Management" and "Suppliers" appear to be largely independent of one another, and both are having major effects on other risk areas. The risk statements in the "Senior Management" risk area have primary or secondary effects on *every* other risk area except "Suppliers." Even though "Sys-

tem Performance" had the largest number of risk statements and the largest number of risk statements identified by the SRE team as "high" in program risk exposure and by the participants as their #1, #2, or #3 top risks, this hierarchy would suggest that these are possibly symptomatic risks, rather than root risks.

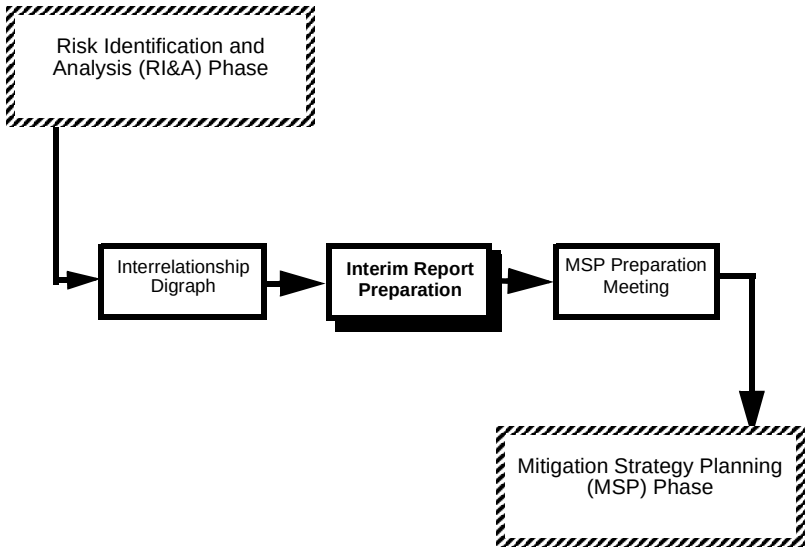
The recommendation in this case would be to mitigate the risk areas in the following order:

1 Senior Management
2 Suppliers
3 Customer Interface

Key Considerations

- Risk areas are only collections of risk statements. The interrelationships must be based on the "condition" element of the underlying risk statements, *not* on the risk area labels.
- Make sure that all team members have the context for the risk statements available during the interrelationship digraph construction phase, and that they refer to it for backup information in cases of disagreement.
- The interrelationship hierarchy will typically be constructed by just one person, most likely the team leader (since the team leader is most personally responsible for the recommendation to the client project manager). The person who constructs it should check back with team members to secure their agreement with the depiction, however.

Interim Report Preparation



Objectives

- to present the results of the Risk Identification and Analysis (RI&A) phase to the project manager in report form
- to recommend which risk areas should be addressed in mitigation strategy planning (MSP) sessions

Who prepares the report?

The SRE team leader is the overall editor of the interim report. This person assigns the preparation of specific subsections of the report to team members, edits the pieces to give the complete report a coherent perspective and a single "voice," and prepares and signs the cover letter for the report.

Timing of Publication

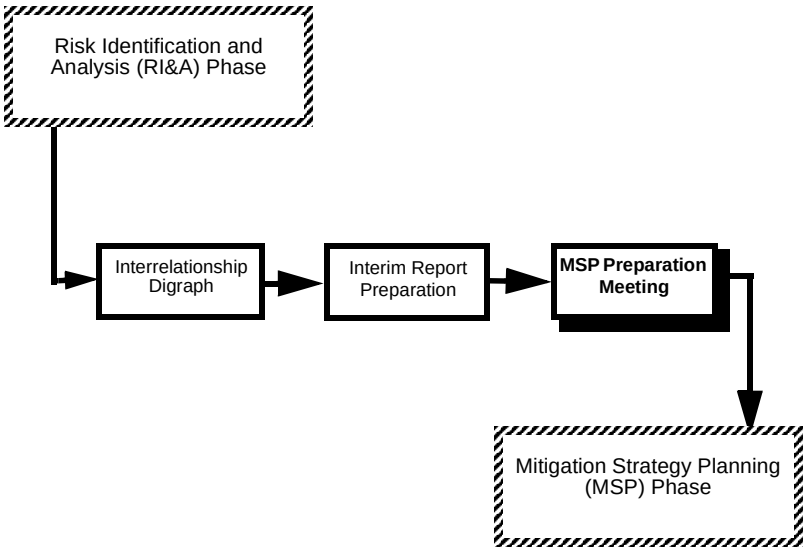
It is important that the interim report be completed quickly, while the enthusiasm for risk management generated by the RI&A phase remains high. Generally, this means that the interim report should be in the client project manager's hands within two calendar weeks of the data confirmation briefing.

Interim Report Outline

An example outline for the interim report follows.

Item	Description
Executive Summary	• summary of risk findings and risk areas • near-term recommendations (“bleeders to be stopped”) • observed strengths (optional—always good for public relations, though)
Introduction	• “caveats” (e.g., “This deals only with risk statements that came out in the interview—it is not an *independent* identification of risks to the project;” “We may not have the technical expertise on the team to evaluate the area in detail;” “This is only a snapshot in time—conditions can change quickly.”) • layout of this report (how to read it)
SRE Process Overview	• shows the larger context into which this RI&A effort fits
Background	• SRE objectives • SRE team makeup • review of the RI&A method used
Findings	• risks by area (include listings of the risk statements in each area) • high-level mitigation recommendations by area (the “low-hanging fruit”) • interrelationship of risk areas, presenting the interrelationship hierarchy and recommending the specific two or three risk areas to be addressed in mitigation strategy planning (MSP)
Conclusion	• next steps • timing of MSP planning meeting
Appendices	• RI&A schedule • (optional) data confirmation briefing slides • (optional) slides from the RI&A phase opening briefing

MSP Preparation Meeting



Objectives

- to meet with the project manager to prepare for the mitigation strategy planning (MSP) activities
- to determine which risk areas will be addressed during the MSP Sessions

Who's in the Room?

- project manager (PM)
- any other project members the project manager chooses to invite (for more technical and managerial knowledge)
- SRE team leader
- SRE team members who will participate in MSP activities

Duration

one hour

Preparation

The following must be completed prior to the MSP meeting.

- Prioritize the list of risk areas generated during the RI&A phase.
- Determine those risk areas that the project is responsible for and can mitigate.

- Determine if a MSP Session is required and beneficial for a particular risk area.
- Propose the number of MSP Sessions to be conducted.

Some or all of these may have been completed as part of the Interim Report.

Approach

The SRE team leader conducts the MSP meeting. The agenda for the meeting should break down as follows:

Review the Interim Report:

- Validate the findings.
- Answer any questions.

Review the SRE team's prioritized list of risk areas for mitigation:

- Review the results of the analysis and prioritization activities from the interim report.

Agree on mitigation areas

- Select mitigation areas to deal with in MSP Sessions.
- Assign the responsible project individual for each mitigation area. (This project member will be responsible for executing the resulting mitigation plan.)
- Assign other project personnel to each selected mitigation area.
- Agree on a schedule for the MSP Sessions (who, when, times, preparation, etc.).

Determine the mitigation goals for the project manager:

- The team leader/facilitator asks the project manager to specify the goals/constraints/interests for mitigating the selected areas.

Set up the distribution of read-ahead material:

- Identify any material or other information that would be beneficial for the session participants.
- Review the logistics for the MSP Session with the on-site coordinator.

Results

- Mitigation areas are agreed upon.
- The project manager's mitigation goals are defined.
- Project personnel is assigned to "lead" mitigation areas.
- The schedule for MSP Sessions is finalized.

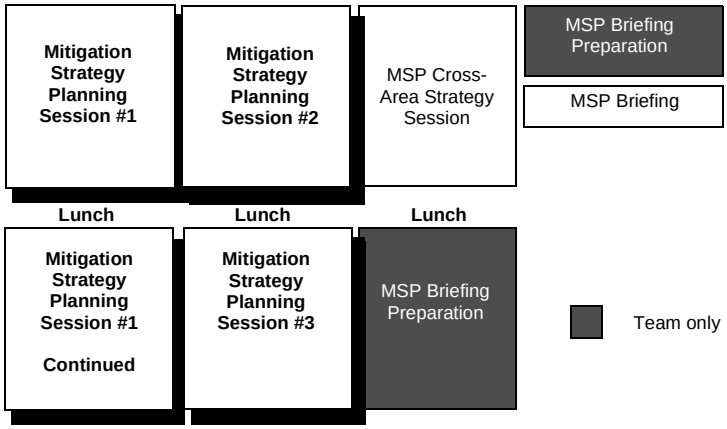
Key Considerations

- The planning meeting is considered informal. However, an optional briefing addressing the results of the SRE team's analysis and prioritization activities may be prepared.
- The construction of the on-site MSP Session schedule may require the SRE team leader to redefine the SRE task:
  - number of sessions
  - additional resources for MSP
  - other SEI (or non-SEI) skills required

Logistics

The MSP meeting is scheduled after the delivery of the interim report. Adequate time should be allowed for the client's review of the report and for the SRE team to prepare for the meeting. The MSP meeting is typically held one to two weeks prior to the start of on-site MSP activities.

Mitigation Strategy Planning Session



MSP Sessions may range from 1/2 to 1 day in length, depending on the size of the risk area and the order of sessions. The first session will take the longest.

Objectives

- to develop a mitigation plan for the risk area, especially those risk statements ranked among the most important to the project by the SRE team or the participants during the RI&A phase
- to identify metrics to track risk and mitigation plan progress
- to teach clients a process and methods for mitigating the rest of their risks

Who's in the Room?

- leader of the client project who is responsible for completing the mitigation area ("owns the risk")
- facilitator of the SRE team
- keeper of context for the SRE team
- domain expert of the SRE team (optional)

NOTE: The SRE team leader is often the facilitator, but not necessarily. The team leader *per se* does not have a role in this session.

Duration MSP Sessions may range from four to eight hours in length depending upon material.

Preparation The following things must be completed prior to conducting an MSP Session:

- The program manager's mitigation goals are defined.
- "Hip pocket" approaches are developed by the SRE team.
- Roles are assigned for facilitator and the team member responsible for context. A domain expert may be requested by the project, and would be a member of the SRE team.
- The medium for capturing plan components is selected (e.g., flipchart and marker).

Approach The facilitator conducts the MSP Session and captures the components of the mitigation plan in front of the participants. A four-hour MSP Session should break down as follows:

Opening the Session: 15 minutes

- Welcome participants.
- Make introductions.
- The client project leader sets expectations about the session results.
- Provide an overview of the MSP Session activities.
- Review the handout material. This should include the "Picture of Success" used for the RI&A phase and all the original risk statements grouped into the risk area.
- Revise or refine the "Picture of Success," if it no longer is persuasive to the participants.
- Resolve any questions/issues.

Identifying Causes: 30 minutes

- The participants review major risks and suspected causes and jot down key or root causes.
- Participants identify their most important key causes until the key causes are exhausted.

- Record the key causes on flipcharts.
- Capture the key causes in a word processing program.
- Reach consensus on a subset of the key causes which the mitigation plan should address.

Identifying Mitigation Goals: 15 minutes

- Review the tentative goals.
- Review the program manager's mitigation goals.
- Modify, delete, or add new goals as necessary.
- Record the goals on flipcharts.
- Capture the goals in a word processing program. One helpful approach for this is to begin each goal statement with "To <verb> ...". {Example: "To increase employee incentives for staying with the company."]
- Reach consensus on the mitigation goals.

Identifying Mitigation Strategies

- Brainstorm and discuss possible strategies. These will define the general approaches to be taken to reach the stated goal. They will typically start with a broad action verb like "Establish," "Research," or "Investigate." [Example that goes with the goal above: "Establish a team to review standard industry benefits for employees in the IT field and make recommendations to the CEO on potential company improvements."]
- Evaluate proposed strategies and reduce them to the desired set.
- Record the strategies on flipcharts.
- Capture the strategies in a word processing program.
- Reach consensus on the mitigation strategies.

Participant Break: 10 minutes

Identifying Mitigation Activities: 65 minutes

- Brainstorm and discuss possible activities for each strategy. Mitigation activities identify *how* the strategies are carried out, and *by whom*. They should also include a *deadline for completion*. They will typically begin with a succinct and specific action verb such as "Complete," "Publish," "Collect," or "Present." [Examples to go with the mitigation activity above: "Complete a charter for the CEO's signature that will establish an employee benefits improvement team—J. Brown—6/5/1999" and "Publish a request for volunteers to serve on the employee benefits improvement team—F. Jones—7/1/1999"]
- Record the activities on flipcharts.
- Capture the activities in a word processing program.
- Reach consensus on the mitigation activities.

Participant Break: 10 minutes

- Print out the goals, strategies, and actions and distribute them to participants.

Identifying Key Measures

- Brainstorm and discuss key measures.

Note: a key measure may be an ongoing measure such as tracking planned vs. actual numbers or it may be a milestone such as the sign-off of an integrated test plan.

- Record the key measures on flipcharts.
- Capture the key measures in a word processing program.
- Reach consensus on the key measures.

Estimating the Scope of Effort

- Divide the participants and team members into as many subteams as there are mitigation strategies.
- Assign each subteam to a mitigation strategy.

- Develop the following estimates for each activity:
  - the number of people involved
  - the number of person-days effort per person
  - the number of calendar days or weeks to complete
- Review the estimates with the entire group and modify as necessary.

Note: If there are a small number of strategies, the entire team can develop estimates for all of the strategies.

- Record the estimates on flipcharts.
- Capture the estimates in a word processing program.
- Reach consensus on the estimates.

Review and Close-out of the MSP Session

- Ensure that all critical or top N risks and mitigation goals are addressed by the selected strategies and activities.
- Mark any corrections.
- Review the strategies and activities for any new risks that may be generated by them. Capture these as standard condition-consequence risk statements on a flip chart for possible later inclusion in the project's risk database.
- Remind participants of the MSP Results Briefing.
- Remind selected participants of the Cross-Area Strategy Session.
- Answer any questions.
- Thank participants for their involvement.

Results

- bulleted list of key or root causes
- bulleted list of mitigation goals (~two to four)
- numbered list of mitigation strategies (~three to five)
- numbered list of mitigation activities (~two to five) for each strategy
- bulleted list of key measures (~three to five)
- an estimate (of people, person-days, and days/weeks) for each activity associated with a given strategy

An electronic version of the flipcharts generated during the MSP Session is sufficient for use in the Cross-Area Strategy Session. However, the

SRE team must complete the documentation of results (for incorporation into the final report and as an artifact to be used by the client project member responsible for the mitigation area). The SRE team should conduct the following activities offline:

- Review and edit the documentation for correctness and completeness (make any necessary adjustments to schedule, resources, actions, etc.).
- Identify any steps that are required to make this an implementable plan.
- Assign appropriate personnel.
- Assign tasks to personnel.
- Obtain approval of the plan.
- Document the results.

Points to Remember

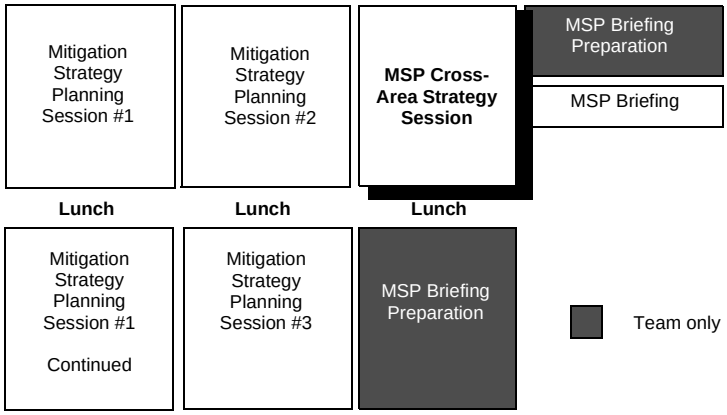
- The depth of planning in an MSP Session varies based on the following:
  - mitigation area scope and criticality
  - client maturity
  - need for mitigation vs. “problem solving”
- Be prepared to renegotiate or extend the session schedule. The team should not cut an area or topic short simply to adhere to the proposed schedule.
- The project will need to further break down the activities into tasks in order to estimate the true effort required, resource allocations needed, and schedule. Realistic estimates can be determined only after sizing the tasks to be performed and the resources that are available to implement them. Estimates developed during the sessions should be used as a guide and starting point by the individuals responsible for implementing the plan.

It is recommended that the final documentation of plans not be conducted until the conclusion of on-site activities. The outcome of the Cross-Area Strategy Session may result in changes to individual mitigation plans.

Logistics

- It is important that the participants be able to see what the facilitator is writing.
- If possible, keep all plan components visible to the participants.
- Each strategy and action developed for a given risk area should have a unique numerical designator.
- If possible, the tool operator should also enter plan components into a briefing slide template. This will assist in the preparation of the MSP Results Briefing.
- Access to copy machines, computers, and printers will keep the activity running smoothly.

Cross-Area Strategy Session



MSP Sessions may range from 1/2 to 1 day in length, depending on the size of the risk area and the order of sessions. The first session will take the longest.

Objectives

- to identify conflicts and synergy among the strategies and actions developed for each mitigation area
- to prioritize mitigation plans and actions
- to teach clients a process and methods for mitigating the rest of their risks

Who's in the Room?

- Client project personnel who are representatives from each mitigation session - ideally all of the mitigation area leaders
- facilitator of the SRE team
- keeper of context for the SRE team
- domain expert of the SRE team (optional)

NOTE: The SRE team leader is often the facilitator, but not necessarily. The team leader *per se* does not have a role in this session.

Duration Typically, a four-hour session is sufficient to review all mitigation plans. However, if a large number of mitigation areas were addressed through the use of parallel sessions, additional time may be required.

Preparation The following things must be completed prior to conducting a Cross-Area Strategy Session:

- All MSP Sessions are complete.
- Mitigation area plans are updated and available for review.
- Team roles are assigned for facilitator, mitigation area representatives, and the team member responsible for context.
- The medium for capturing plan components is selected (e.g., flipchart and marker).

Approach The SRE team facilitator conducts the Cross-Area Session and captures the identified conflicts and synergy in front of the participants. A four-hour Cross-Area Session should break down as follows:

Opening the Session: 10 minutes

- Welcome participants.
- Make introductions.
- The facilitator sets expectations about the session results.
- Provide an overview of the Cross-Area Session activities.
- Review the handout material.
- Resolve any questions/issues.

Review Mitigation Area Results: 60 minutes

- Each plan is reviewed by the mitigation area representatives.
- Make each plan visible to all participants (hang flipcharts on wall).

Participant Break: 10 minutes

Identify Conflicts, Commonalities, Dependencies, and Possible Sequencing: 75 minutes

- Identify any conflicts (strategies or actions that are in disagreement with each other).
- Identify any commonalities (similarities in strategies and actions that suggest a combination or deletion for the sake of efficiency).
- Identify any dependencies (when a particular activity can not begin until another has completed).
- Record conflicts, commonalities, and dependencies on flipcharts.
- Capture conflicts, commonalities, and dependencies in a word processing program.
- Update individual mitigation plans as required.

Participant Break: 10 minutes

Resolve Conflicts: 45 minutes

- If applicable (and possible), resolve any identified conflicts.
- Revise, add, or eliminate actions as needed.
- Review the impact to a mitigation area whenever changes are made to the area's action.
- Record any resolutions on flipcharts.
- Capture any resolutions in a word processing program.
- Update individual plans to reflect conflict resolution (or need for future consideration).

Prioritizing Strategies and Actions: 30 minutes

- Determine the order of execution for strategies and actions considering the following
  - the contribution of strategies and actions to mitigation goals
  - costs
  - dependencies
- Record the prioritized list on flipcharts.
- Capture the prioritized list in a word processing program.

Review and Close Out Cross-Area Session: 10 minutes

- Remind participants of the MSP Results Briefing
- Answer any questions.
- Thank participants for their involvement.

Document Overall Mitigation Plan: offline

- Document the results of the MSP and Cross-Area Sessions.

If the updating of individual plans and documenting of overall plan can not be accomplished during the Cross-Area Session, team members can be assigned to complete these tasks offline or in parallel with the MSP results preparation activities.

Results

- Mitigation strategy and action conflicts are resolved.
- Individual mitigation plans corrected and updated.
- Mitigation strategies and actions are prioritized
- The overall mitigation plan is documented and includes the following:
  - prioritized list of strategies and actions
  - unresolved conflicts
  - dependency or relationship graph/matrix
  - electronic plan charts updated for use in MSP Results Briefing

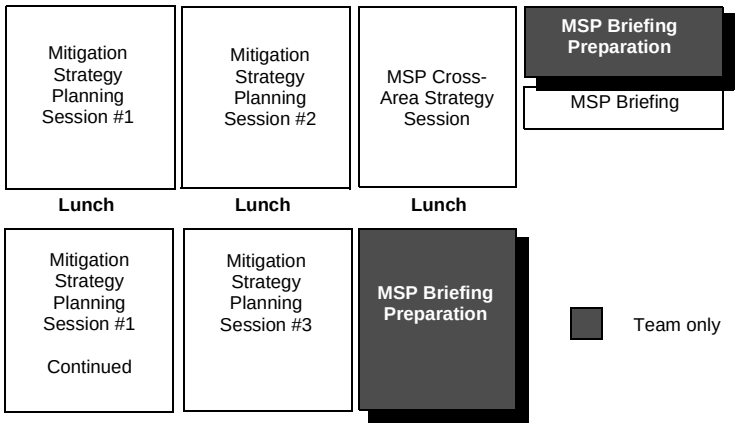
Points to Remember

- The Cross-Area Strategy Session is considered to be an optional activity and the session may be unnecessary if the same personnel participated in all MSP Sessions or if the mitigation areas are so disjointed they don't overlap in strategies and actions.
- Even if it appears that a Cross-Area Strategy Session is not required, the team should consider the following:
  - All mitigation plans should be reviewed quickly for potential conflicts and synergy.
  - Mitigation area prioritization (resulting from MSP meeting) should be revisited at the conclusion of the MSP Sessions.

Logistics

- It is important that the participants be able to see what the facilitator is writing.
- If possible, keep all plan components visible to the participants.
- Access to copy machines, computers, and printers will keep the activity running smoothly.

MSP Briefing Preparation



MSP Sessions may range from 1/2 to 1 day in length, depending on the size of the risk area and the order of sessions. The first session will take the longest.

Objectives

- to update and finalize the mitigation plans developed in the Mitigation Strategy Planning sessions
- to create the MSP Results Briefing presentation materials

Who's in the Room?

SRE team

Duration

4 - 5 hours

Preparation

The following must be accomplished prior to creating the MSP Results Briefing:

- All MSP Sessions are complete.
- Cross-area strategy session is complete.
- Mitigation area plans are updated and complete.
- Consideration of project's next steps have been made.

Approach

The team leader leads the team in developing the MSP Results Briefing presentation. The presentation should include the following:

- "boilerplate" cover page
- SRE objectives
- review of the RI&A phase
- review of off-site analysis conducted prior to MSP
- MSP process review
- mitigation plans
- description of "next steps" for the project and the SEI
- summary
- placeholder for project manager's closing comments

After the presentation has been created, do the following:

- Make transparencies of slides.
- Make a hard copy of the slides for the project manager.
- Make a dry run of the presentation.
- Make hard and soft copies of the mitigation plans for the responsible project personnel.

Results

The results are the following Results Briefing presentation materials:

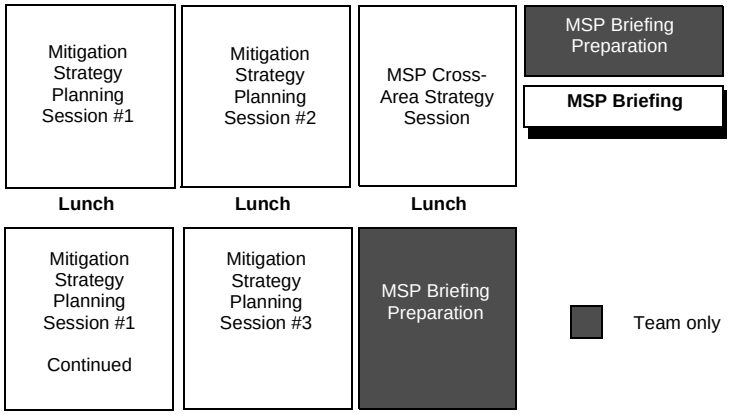
- transparencies of slides
- hard copy of slides for the project manager
- hard and soft copies of the mitigation plans

Key Considerations

- The MSP Results Briefing is the presentation during which all MSP participants see how their own planning efforts contributed to the overall mitigation plan. More importantly, all project personnel will have an opportunity to see how the top risks from the risk identification and analysis activity will be addressed and in what order. They again "buy in" to the process, by seeing that their risks were captured and are being addressed in a proactive manner.
- Encourage all participants to attend the MSP Results briefing.

Logistics It is best to have a direct display device to make this presentation directly from the slide presentation software. If this is not possible, quick access to a photocopier for creating transparencies and making a hard copy for the project manager becomes essential.

MSP Results Briefing



MSP Sessions may range from 1/2 to 1 day in length, depending on the size of the risk area and the order of sessions. The first session will take the longest.

Objective

to present the project with the results of the mitigation strategy planning (MSP) activity

Who's in the Room?

- project manager
- All MSP participants
- Any other project members the project manager chooses to invite
- SRE team

Duration

one hour

Preparation

Prior to giving the MSP Results Briefing, the following must be accomplished:

- Both hard and soft copies of developed mitigation plans have been prepared.

- Presentation transparencies and a hard copy of them have been prepared for the project manager.
- The project manager and all participants are in attendance.

Approach

The presentation is a formal briefing. The following will occur:

- The project manager will introduce the SRE team leader.
- The team leader will present the MSP Results Briefing.
- After the presentation, the team leader invites the project manager to comment.
- The project manager shares comments with the audience.
- The team leader gives a hard copy of the presentation to the project manager.
- The team leader gives copies of mitigation plans to the participants who are responsible for the mitigation area.

Results

official ending to the on-site MSP activity

Key Considerations

- The project manager and participants need to see a coherent and focused picture of the results. The briefing includes a section on the next steps - where the program needs to go from here with the developed mitigation strategies and actions. This area needs to be discussed with the project manager and the project manager's representatives when the results of the MSP Sessions are presented. The project manager needs to understand that action on the MSP Session results can begin immediately.
- Participants need to see their manager introduce the team leader at the beginning of and summarize the importance of the risk management activity to the project at the end of the briefing.
- The MSP Results Briefing is a tangible result of the on-site MSP activities. Take time to prepare the words as well as the briefing slides.

SRE Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Index

U